A Machine Learning Model for Revenue Forecasting at Local Government Authorities in Kenya: A Case of Nairobi City County

Waweru Gichuhi

090700

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Information Technology at Strathmore University

School of Computing & Engineering Sciences

Strathmore University

Nairobi, Kenya

June, 2025

Declaration and Approval

Declaration

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

Student's Name: Waweru Gichuhi

Sign: _____ Date: _____22nd May 2025_____

Approval

The thesis of Waweru Gichuhi was reviewed and approved by the following:

Prof. Bernard Shibwabo

Associate Professor, School of Computing & Engineering Sciences,

Strathmore University

Dr. Julius Butime,

Dean, School of Computing & Engineering Sciences,

Strathmore University

Prof. Bernard Shibwabo,

Director of Graduate Studies,

Strathmore University

Abstract

Revenue forecasting is a key component of the fiscal cycle in local governments. It enables officials to estimate how much revenue they can expect to raise from various revenue sources under their mandate and therefore, what potential they have to meet their annual expenditure requirements. Budgets that are drawn up with poorly forecasted revenue figures are prone to inevitable shortfalls, and consequently, the inability to honor obligations to ongoing development projects, recurrent expenses and service delivery to citizens. Expert judgement is commonly used in forecasting revenue due to its simplistic nature. However, this is not always a reliable method, nor is it backed by factual evidence that may be used to incorporate various other aspects that also influence changes in revenue potential. The objective of this research was to develop a machine-learning (ML) model for revenue forecasting at local government authorities in Kenya. The study adopted an experimental research design to analyze the change in revenue estimates over time. Non-probability sampling was used to review primary data from Nairobi City County, and thereafter, one statistical and six different machine-learning models were evaluated on the same data set. The statistical model used was SARIMAX, which was informed by literature and was used as a baseline to measure the efficacy of an AI approach. The ML models comprised of regressors, tree-based algorithms and a recurrent neural network. They were trained on 14 different revenue streams spanning over 36 months, validated on 9 months of collections and tested over a 6-month spread. The statistical model registered a forecast accuracy of 53.9% on the total revenue collected, while the highest accuracy score on a single revenue stream was attained by the Random Forest model at 79.8% on Land Rates collections. The best overall ML model across all revenue streams was found to be the Ridge Regressor. These ML results outperformed those of the benchmark model, and showed the potential that machine learning has in budgeting, revenue forecasting and decision-making processes at the county level.

Keywords: Machine Learning, Revenue Forecasting, Local Government, Time Series Analysis

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AI	-	Artificial Intelligence
ANN	-	Artificial Neural Network
ARIMA	-	Autoregressive Integrated Moving Average
CRA	-	Commission on Revenue Allocation
IMF	-	International Monetary Fund
KNN	-	K-Nearest Neighbor
LGA	-	Local Government Authority
LGBM	-	Light Gradient-Boosting Machine
ML	-	Machine Learning
MLR	-	Multiple Linear Regression
MTBF	-	Medium-Term Budgetary Framework
NCC	-	Nairobi City County
OCOB	-	Office of the Controller of Budget
OSR	-	Own Source Revenue
PFM	-	Public Finance Management
RAD	-	Rapid Application Development
RF	-	Random Forest
RMS	-	Revenue Management System
RNN	-	Recurrent Neural Network
SARIMAX	-	Seasonal ARIMA with Exogenous Regressors
SVM	-	Support Vector Machine
SVR	-	Support Vector Regression
TSA	-	Time Series Analysis

Acknowledgements

I am forever grateful to God for His providence and grace, and for granting me the opportunity to follow through on this noble pursuit. I acknowledge my family members, Dr. and Mrs. Gichuhi, Njeri and Neema, and my partner Juliet, for their constant encouragement and support in my journey. I wish to thank my research supervisor, Prof. Bernard Shibwabo for his patience, guidance and mentorship in fulfilling all academic requirements. My gratitude goes out to staff members of the Nairobi City County Government's Revenue and ICT Departments, as well as faculty staff at the School of Computing and Engineering Sciences, for their assistance during my studies. Finally, I appreciate my fellow comrades for the friendship and memories.

Definition of Terms

Local Government Authority	One of the decentralized, sub-national spheres of government distributed across the territory of a country to run/regulate political, fiscal and administrative government functions or public services on their own (United Nations, 2018).
Machine Learning	Machine learning is the study of computer algorithms that allow computer programs to automatically improve through experience (Mitchell, 1997).
Medium-Term Budgetary Framework	Institutional policy instruments that allow the extension of the horizon for fiscal policy-making beyond the annual budgetary calendar (Sherwood, 2015).
Predictive Analytics	The use of data, statistical algorithms and machine learning techniques to identify the likelihood of future outcomes based on historical data (SAS, 2023).
Public Finance Management	The set of laws, rules, systems and processes used by sovereign nations and sub-national governments, to mobilize revenue, allocate public funds, undertake public spending, account for funds and audit results (Lawson, 2015).
Revenue Forecast	A fiscal management tool that presents estimated information based on past, current, and projected financial conditions (Government Finance Officers Association, 2023).

1. Introduction

1.1. Background

Revenue forecasts may be defined as "estimates of what governments will collect from various sources, such as income taxes, value-added taxes, corporate taxes, and excises, which together determine the funds available to allocate to various public programs" (McCurdy, Galper, & Raza, 2019; Fikriya & Hikmawati, 2020). There is a delicate interdependency between national and local government in that budgetary shortfalls can directly hinder the counties' ability to deliver essential services to its citizens. If governments spend more than they collect, this may lead to borrowing money or raising taxes to finance deficits, which can have long-term negative effects on the economy (Jenkins, Kuo, & Shukla, 2000; Onu & Ezeji, 2017).

The promulgation of the Kenyan constitution in the year 2010, paved way for the establishment of devolved government units by creating forty-seven counties modeled after the local government authorities previously identified as Districts. This devolution manifests in both administrative and fiscal dimensions. From a fiscal sense, decentralization refers to "the transfer of financial authority to the sub-national governments by reducing the conditions on the intergovernmental fiscal transfer of resources and granting sub-national units greater authority to generate their own revenue" (Mwenda, 2010; Centre for Economic Governance, 2018).

Counties, by law, therefore have a mandate to generate revenue from local sources in order to finance their annual budgetary expenses. This is in addition to funds allocated from central government, whose factors are determined by the Commission on Revenue Allocation (CRA) and which are disbursed through National Treasury, in order to meet fiscal obligations such as recurrent expenditure and development projects at the county level (Commission on Revenue Allocation, 2018; Office of the Controller of Budget, 2022).

According to the Centre for Economic Governance (2018), thirty-four County Governments in Kenya have taken steps to automate their revenue collection and management processes in order to enhance own source revenue generation and raise funds to deliver essential services to citizens. The body also notes that, "97% of these Counties have only been able to utilize the Revenue Collection module, and no other processes from the Public Financial Management Act (2012)". These are illustrated in Figure 1.1.

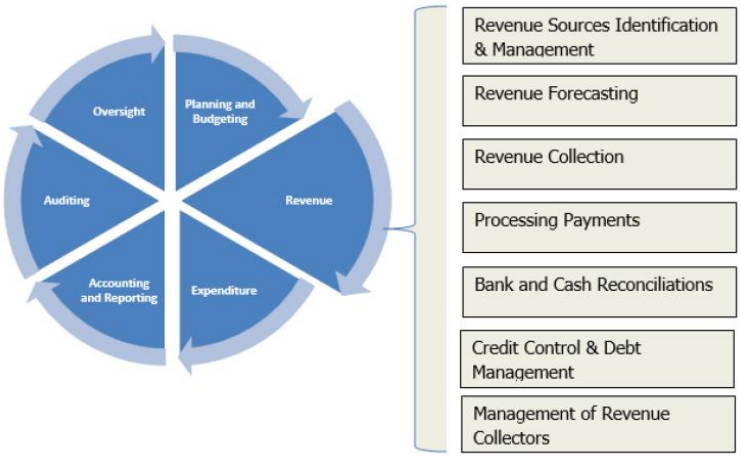


Figure 1.1: Revenue collection processes from the Public Financial Management Act (2012) (Adapted from Centre for Economic Governance, 2018)

There are several methods used in performing revenue forecasting, which fall into two broad categories of qualitative and quantitative methods. Qualitative (or subjective) methods rely on expert judgment and opinions to forecast tax revenues. An example of subjective revenue forecasting is through consensus forecasting, whereby information from several forecasts is aggregated to arrive at a final revenue estimation (Rubin, Mantell, & Pagano, 1999). Quantitative methods, as Chindia, Wainaina, Kibera and Pokhariyal (2014) note, use historical data to establish relationships and trends that can be projected into the future. Whereas current methods are descriptive in nature, there is an opportunity to use emerging technologies like machine learning to predict the best solutions to complex challenges by measuring and analyzing the uncertainty that decision makers must deal with (Talluri & Ryzin, 2014; Yang, He, & Jiang, 2018).

1.2. Problem Statement

Revenues at County Governments in Kenya fall into two main categories, those collected directly from residents (Own Source Revenues) and those collected at the national level and passed on to Counties as inter-governmental transfers (Agile and Harmonized Assistance for Devolved Institutions (AHADI), 2020). CRA notes that within these Own Source Revenues; ICT plays an important role in the automation of revenue collection and management processes, thereby enhancing efficiency and transparency (Commission on Revenue Allocation, 2018). Revenue forecasting is one of the key tenets of this automation process,

considering that existing systems comprise of transactional databases with records of collections done per revenue stream at counties.

Most Revenue Management Systems (RMS) in use have inadequate forecasting capabilities built into them which is disadvantageous to the County governments, who need such insights to plan and optimize their resources each fiscal year. Indeed, the IMF has emphasized that in order to promote fiscal transparency and prevent ad hoc cuts to public investment and other high priority expenditures, it is crucial to have accurate income estimates (International Monetary Fund, 2018). Onu and Ezeji (2017) also concur and observe that poor forecasting eventually means government projects cost more, run behind schedule or produce fewer benefits than expected. Poor budgets also lead to over or under expenditure and missed opportunities to spend on other worthwhile projects. Ultimately, proper forecasts inform the institution's medium-term budgetary framework (MTBF), which should extend beyond an annual cycle to at least a three-year rolling outlook, depending on the term of an elected government. This helps avert a strain on public finances and considerable sustainability risks (Sherwood, 2015).

There are three main approaches to predicting municipal revenue. These are judgmental (expert) methods, extrapolative (trend) methods, and deterministic (econometric) methods (Hajek & Olej, 2010; Reddick, 2004). However, revenue sources are increasingly getting complex, especially since the projected revenue and its input factors may not always be linearly correlated (Hajek & Olej, 2010). This research proposed a scientific approach to addressing this issue by using machine learning models to perform revenue forecasting, and helping to support decision-making processes at the County level. With such a solution, the counties can better anticipate the performance of their revenue streams and adjust their budgets to suit their cash flow.

1.3. Research Objectives

1.3.1. General Objective

The purpose of the research was to integrate a machine-learning model into the revenue forecasting process at local government authorities in Kenya. This would help increase the accuracy of revenue forecasts at the county government under study, and the results obtained would be instrumental to the county's budgeting cycle.

1.3.2. Specific Objectives

i) To investigate the challenges associated with revenue forecasting
ii) To evaluate methods, techniques and approaches used in forecasting revenue collection
iii) To formulate a model to forecast revenue collection using machine learning algorithms
iv) To validate the proposed model

1.4. Research Questions

i) What are the challenges associated with revenue forecasting?
ii) Which methods, techniques and approaches are used to forecast revenue collection?
iii) How can machine-learning algorithms be used to formulate a model to forecast revenue collection?
iv) How can the model be validated?

1.5. Justification

In principle, accurate local government revenue forecasting can assist governments and major agencies in making informed decisions. (Yang, He, & Jiang, 2018). Business intelligence may be used by governments to improve local tax revenue (Fikriya & Hikmawati, 2020). Chung, Williams, and Do (2022) also note that an advantage of machine learning algorithms is that they boost the effectiveness of existing amenities by precisely anticipating demand or targets.

This research explored predictive, machine learning approaches to forecast revenue collection. This would in turn help local government authorities to better identify growth potential across various revenue streams based on analyses of revenue collection trends, and support decision-making when adjusting revenue targets for each subsequent fiscal period. Indeed, according to Batóg and Batóg (2021), feasible revenue and expense projections can give a sense of the availability of funds in the institution, analyze financial risk, determine whether or not services can be maintained, and pinpoint the major factors that affect the quantity of revenue.

1.6. Scope

Whereas automation of revenue collection is seen as a major contributor to netting higher inflows and revenue target compliance at local government authorities, several other interventions are needed to have a predictable basis for budgeting (Institute of Public Finance - Kenya, 2020). This research aims to extend Revenue Management System automation initiatives that have already yielded historical year-on-year collection values, by applying machine-learning models to forecast expected revenue growth or decline in those jurisdictions. Multiple variables were assessed to determine those that directly influence the model's results and ultimately the decision-making process.

1.7. Limitations

The PFM Act of 2012 outlines both revenue and expenditure as key pillars of public accounts and fiscal prudence. Whereas the two are intimately linked, the study was limited only to the revenue collection process, and further still, only to own source revenue (OSR). Other monetary sources like equitable share from national government, conditional grants, appropriation in aid and facility improvement fund were not considered due to complexities in how these revenues are obtained and incongruent records on the revenue management system.

Sensitivity of information that was not in the public domain also posed a challenge when collecting data from the sampled county. This was especially true after widespread resistance and chaos was witnessed from the public following the voting in of the Finance Bill 2024/2025 at national government level. The researcher relied on making formal requests to concerned parties while leveraging strategic networks to access historical revenue collection figures, while giving proper justification for their use.

2. Literature Review

2.1. Introduction

Forecasts are useful tools for assessing the viability of an annual budget, the effects of policy decisions based on availability of funds, and extrapolating the generation of different revenues and their consumption through expenditure (Swanson, 2008). According to Williams & Kavanagh (2016), governments use revenue projections in budgeting because identifying resource availability and constraints is an element of this planning process. They further explain that forecasting helps governments compare their actual vs. budgeted figures during any given fiscal period. Forecasting also helps to accurately plan future expenditures, optimize the allocation of resources, and consequently guarantee and enhance people's quality of life. (Yang, He, & Jiang, 2018; Hajek & Olej, 2010).

Local governments set targets to encourage revenue collection. A target that is set as a simple adjustment of the previous year's revenue target may not capture the true potential for revenue collection; therefore, prediction becomes a necessary tool to reveal this potential (Fikriya & Hikmawati, 2020). Swanson (2008) puts forward the idea that baseline forecasts are based on recurring revenues and expenditures that are at least five years out. In order to establish a solid forecast, one must consider several factors that influence revenue collection at the local government level. Table 2.1 shows some of the attributes that may be evaluated.

Table 2.1: Factors that influence revenue (Adapted from Swanson, 2008)

Exhibit 1: Factors that Influence Revenue

Revenue Source, Type	Influence Factors
Property Tax	Impact from housing market, home sales,foreclosure activity, assessed valuation trend, population, personal income, unemployment rate
Property Transfer Tax	See Property Tax
Sales Tax	Retail/commercial activity, consumer trends, impact of housing market, population growth, impact of regional economic forces, planned developments, personal income, unemployment rate
Franchise Tax	Population change, general demand, new development, per unit usage, potential rate increases (external providers and city rates)
Transient Occupancy Tax	Hotel/motel development, occupancy rate trends, travel & tourism trends
Other Tax	Impact factors affecting other taxes
Intergovernmental—Federal	Availability of federal grants and reimbursements, legislative and budget actions
Intergovernmental—State	For vehicle in lieu fees, similar to factors that affect property and sales taxes; availability of state grants and reimbursements, legislative and budget actions
Licenses & Permits	Similar factors as property and sales taxes
Fines & Forfeitures	Fines and penalty rates; similar to factors that affect property and sales taxes
Charges for Service	Similar to factors that affect taxes, cost of providing services, and rate schedule for cost recovery

2.1.1. Global Revenue Forecasting Trends

Surveys by Engstrom, Ho, Sharkey, and Cuéllar (2020) and studies by Chung, Williams, and Do (2022) indicate that 45% of 142 federal agencies in the United States of America have experience using AI algorithms. This usage cuts across state departments such as Customs and Border Protection, Internal Revenue Service, Food and Drug Administration and the Social Security Administration. Across continents in Indonesia, research by Fikriya and Hikmawati (2020) suggest there is potential for the use of predictive analysis by Local Government to improve local tax revenue and minimize tax reduction. However, there is hardly any documented use of AI and ML techniques when generating revenue forecasts in Kenya. It is still important to appreciate the performance of existing techniques as seen through published county budget forecasts, and to compare how closely these figures match the actual revenue collected during the period under review.

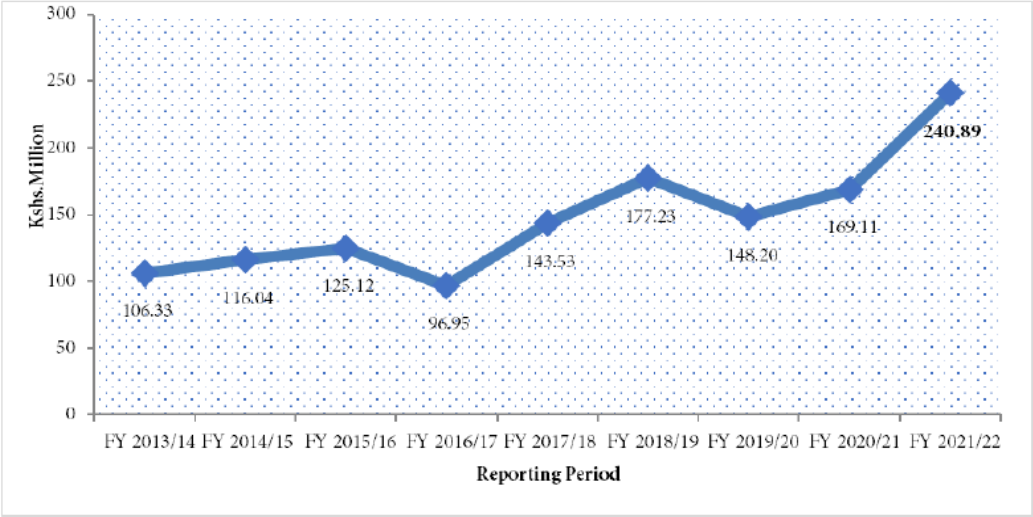
2.1.2. Revenue Forecasting Trends in Kenya

Based on recent reports from Office of the Controller of Budget (2022), the budget performance for all 47 counties in the financial year July 2021 to June 2022 shows some disparity between forecasted estimates and actual own source revenue collected. The report shows that out of a cumulative OSR budget of Kshs.60.42 billion, counties managed to raise Kshs.35.91 billion, which represents an average performance of 59.4% against their targets. In the bottom tenth percentile, we have the following five counties whose annual performance was well below the national average. These are Busia at 30%, Murang'a at 32.9%, Kajiado at 33.1%, Garissa at 43.7% and Embu at 43.8%.

A comparison may be made with the upper ninetieth percentile where only four counties achieved their set annual target. These are Turkana at 113.5%, Migori County at 110.5%, Lamu County at 105.5% and Vihiga at 101.6%. Taking a closer look at some of the best and worst performers reveals more trends in actual OSR collections, which ideally should be the basis upon which future forecasting is done. We may further take keen note of two counties that are from the same region of Western Kenya, as depicted in Table 2.2, and plotted in Figure 2.1 and Figure 2.2.

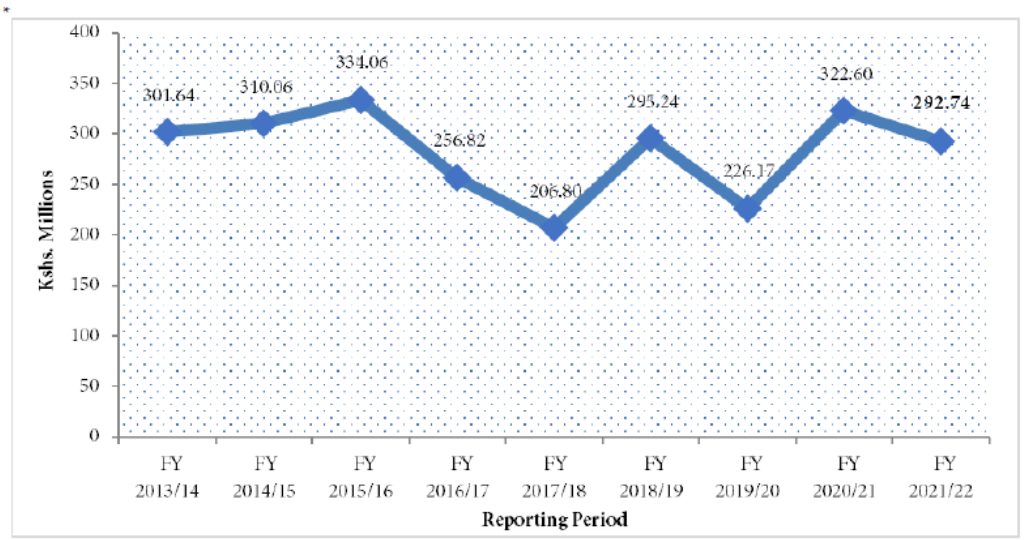
Table 2.2: Busia County and Vihiga County revenue performance in FY 2021/22 (Adapted from OCOB, 2022)

Own Source Revenue	Annual Budget Allocation (in Kshs.)	Actual Receipts in the FY 2021/22 (in Kshs.)	Actual Receipts as Percentage of Annual Allocation (%)
Busia County	976,108,322	292,736,456	30
Vihiga County	232,658,878	240,890,593	103.5



Source: *Vihiga County Treasury*

Figure 2.1: Vihiga County OSR performance from FY 13/14 to FY 21/22 (Adapted from OCOB, 2022)



Source: *Busia County Treasury*

Figure 2.2: Busia County OSR performance from FY 13/14 to FY 21/22 (Adapted from OCOB, 2022)

The report further recommends counties to "review their FY 2022/23 revenue targets, confirm that they are realistic and implement strategies to mobilize their own source revenue collection" (Office of the Controller of Budget, 2022). Some of the challenges noted by Reddick (2004) when forecasting revenue, especially in developing countries, is uncertainty. A conservative approach is used to underestimate expected revenue so as to have a greater chance of a surplus. He also notes that local government officials often lack the knowledge of revenue forecasting techniques like time series, hence the use of qualitative methods.

2.2. Existing Revenue Forecasting Techniques

Forecasting techniques can be classified under two broad categories. The first of these is the qualitative approach, which is subjective and dependent on expert judgement. The second approach consists of quantitative techniques, which are objective and rely upon the use of statistical methods to process historical data (Reddick, 2004; Onu & Ezeji, 2017). Both categories may further be exemplified through different methods used to forecast revenue collection potential.

2.2.1. Qualitative Techniques

2.2.1.1. Naïve: Judgmental or Expert Forecasting

This method relies on the advice of someone who has experience with the revenue source. In a typical local government setup, these individuals would be those conversant with their systems or knowledgeable about supporting evidences (Reddick, 2004). They are able to make judgements or guesses about the trends that can be expected from each category. It is an appropriate method when there is little relevant data available, things are changing quickly, and/or predicted revenues don't follow a steady pattern (CEPAL, 2015).

2.2.2. Quantitative Techniques

2.2.2.1. Incremental: Extrapolative, Trend or Time Series Forecasting

This method uses time series analysis with mathematical formulae to weigh values from a historical series to predict future values. A time series forecast is influenced by four components; the trend, seasonal, cyclical and random noise component. Trend refers to the average direction of change in data over time and may be characterized as linear if it increases or decreases at a constant rate over time; or exponential if it increases by a constant percentage

over time. Seasonality refers to revenue cycles that change within a year in response to some phenomena. Cyclical fluctuations of a time series result from the business cycles of the economy. Finally, random noise or shocks refer to unexpected events that may distort trends that otherwise exist over the long-term. (CEPAL, 2015). The commonly used types of time series forecasting models include the following:

Moving Average model

This is a common technique used to spot trends in a data set as an average of a subset of numbers. It is calculated as a statistical mean of lead (future values) using lag (past values) recorded over a given period. According to Johnston, Boyland, Meadows, and Shale (1999), two variants of this model include the simple moving average, represented by the equation of n values, made after the knowledge of the observation y_t as m, defined as:

$$m = \left(\frac{1}{n}\right)\sum_{i=0}^{n-1} y_{t-i}$$

and the exponentially weighted moving averages represented by the equation:

$$Z_i = \lambda \bar{X}_\iota + (1-\lambda)Z_{i-1}$$

where $Z_0 = 0$ and λ $(0 < \lambda \leq 1)$ is the weighing or smoothing parameter that determines the weighting of past data (Mahmoud & Woodall, 2010).

Exponential Smoothing model

This is a corrected moving average of forecasts that has been revised to account for the inaccuracy seen in earlier forecasts (CEPAL, 2015). A simple exponential smoothing may generate forecasts using the equation:

$$S_t = \alpha X_t + (1-\alpha)S_{t-1} = S_{t-1} + \alpha(X_t - S_{t-1})$$

where α is the smoothing parameter and takes values between 0 and 1 (CEPAL, 2015).

Autoregressive Integrated Moving Average (A.R.I.M.A.) model

This is a time series model in which future values are determined to be linear functions of historical data. It consists of an autoregressive process whereby a random component and a linear combination of prior observations make up each observation. (Fattah, Ezzine, Aman, Moussami, & Lachhab, 2018). This is represented by the equation:

$$Y_t = \alpha_1 Y_{t-1} + \varepsilon_t$$

The integrated process is where the model is affected by the cumulative effect of an activity and can be represented by the equation where the random perturbation ε_t is a white noise.

$$Y_t = Y_{t-1} + \varepsilon_t$$

The moving average process is where the current value is a linear combination of the current disturbance with one or more previous perturbations, and is represented by the equation:

$$Y_t = \varepsilon_t - \theta_1 \varepsilon_{t-1}$$

where Y_t is a linear function of the preceding values, ε is a random shock and α_1 is the self-regression coefficient (Fattah, Ezzine, Aman, Moussami, & Lachhab, 2018).

However, Reddick (2004) contends that these extrapolative methods only predict fluctuations throughout the projection period but fail to pinpoint the factors influencing the forecast, hence the need for causal modeling.

2.2.2.2. Causal: Deterministic and Econometric Forecasting

Causal methods are strategic and look at the long term but they do require more time resources and expenses than simpler time series tools. Deterministic techniques consider other variables, not just time, to project revenues (Reddick, 2004). Hayes (2022) defines Econometrics as "the use of statistical methods to develop theories or test existing hypotheses in economics or finance and to forecast future trends from historical data". Reportedly, it utilizes tools such as frequency distributions, probability and probability distributions, statistical inference, correlation analysis, simple and multiple regression analysis, simultaneous equations models and time series methods.

A general equation for causal regression may be represented as:

$$Y = \beta_0 + \beta_1 X + u$$

where β_0 is the intercept parameter, β_1 the slope parameter, Y represents the dependent variable, X is the dependent variable and u is the identifier for the slope variation (Mwangi, 2016; Campbell & Campbell, 2008).

2.3. Revenue Forecasting Using Machine Learning Algorithms

ML-based forecasting has several advantages over classical methods. Firstly, traditional approaches require significant time and effort to manually engineer features and select model steps, especially where many individual time series are involved (Salinas, Flunkert, & Gasthaus, 2019). Secondly, Exponential Smoothing and ARIMA are shown to work best with linear, autoregressive processes and trail behind the performance of ML models like Light Gradient Boosting Models and Recurrent Neural Networks in handling non-linear patterns from heterogenous data generating processes (Hewamalage, Bergmeir, & Bandara, 2021). Thirdly, ML generalization and accuracy is also better, partly as a result of cross-learning between series, and specifically as the data set size and complexity increases (Makridakis, Spiliotis, & Assimakopoulos, 2022).

According to Montero-Manso & Hyndman (2020), there are two ways to approach forecasting of groups of time series, that is, either by use of global methods or local methods. Global methods fit a single function to all time series in the data set. On the other hand, local methods assume a univariate series and model a regression function for each type of series. Local models are computationally expensive to scale and maintain, and are generally prone to overfitting, especially when dealing with a small sample size (Yingjie & Abolghasemi, 2024). However, global methods can model a single function over related series, especially if their data is generated from a similar process, though this is different from multivariate models that learn correlations between series simultaneously (Rabanser, Januschowski, Flunkert, Salinas, & Gasthaus, 2020). The inherent nature of municipal data is such that own source revenues share similar characteristics like the period of collection (one fiscal year) and extraneous factors like the administrative staff involved, citizenry, policies and events (political processes, COVID-19 etc.), all of which imbue them with similar trends and seasonality. This correlation gravitates them towards the application of global methods of time series forecasting. An examination of commonly cited ML algorithms used in time series forecasting is further carried out to appreciate how they function and achieve their results on regression-based tasks.

2.3.1. Artificial Neural Networks

Neural computing refers to a pattern recognition methodology for machine learning. A Neural Network (NN) can be specified as a logical model, which is designed based on the human brain. The human brain contains interconnected nerve cells called neurons. These neurons are the basic functional unit of the human (or animal) nervous system (Madhu, et al., 2021) as shown in Figure 2.3.

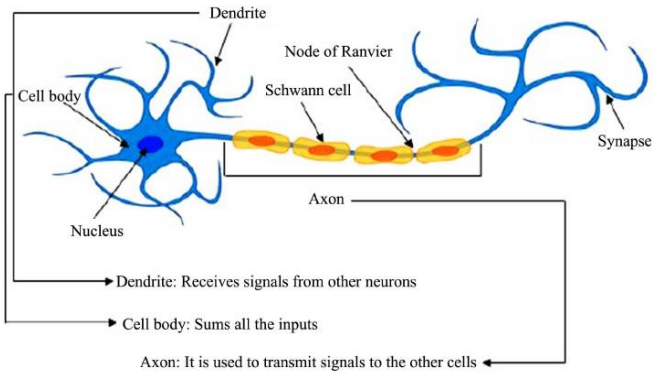


Figure 2.3: A typical neuron with its different parts (Adapted from Madhu, et al., 2021)

Multiple neurons' messages are taken up by a dendrite. To generate input, the cell body, or soma, combines all the incoming signals. A neuron fires when this combination crosses a threshold value, and the signal travels down the axon to the neighboring neurons. (Madhu, et al., 2021). A synapse is situated at the point of interconnection between neurons and can increase or decrease the signal strength between them, causing excitation or inhibition of a subsequent neuron. An Artificial Neural Network (ANN) mimics this biological neural network and borrows several concepts from it such as a node (cell body/soma), input (dendrites), output (axon) and weight (synapse) (Sharda, Delen, & Turban, 2014) as depicted in Figure 2.4.

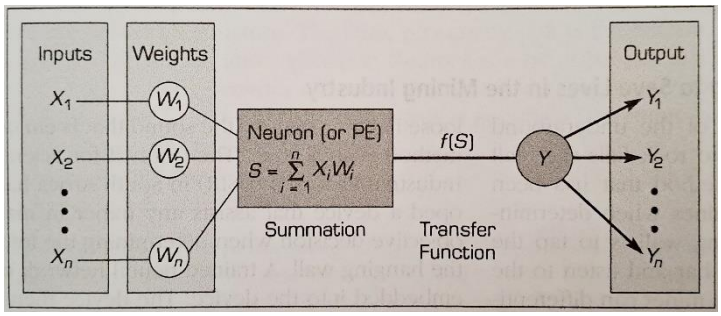


Figure 2.4: Processing information in an Artificial Neuron (Adapted from Sharda, Delen, and Turban, 2014)

2.3.1.1. Elements of an Artificial Neural Network

Three layers make up ANNs and these are the input layer, intermediate (hidden) layer and output layer. According to Hajek and Olej (2010), a feed forward neural network (where information flows in one direction) utilizes layers of interconnected neurons. The output of one neuron gets dispersed into the inputs of neurons in the adjoining layer. Consequently, the input values only move from input to hidden layers and vice versa. This can be seen in Figure 2.5.

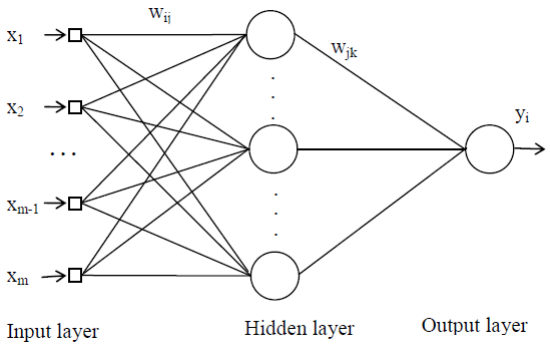


Figure 2.5: General structure of a feed-forward neural network (Adapted from Hajek and Olej, 2010)

A summation function calculates the weighted sums of each processing unit's input components. It adds up all the values for a weighted sum Y by multiplying each input value by its weight (Sharda, Delen, & Turban, 2014). This can be represented in a formula for n inputs in a single neuron as:

$$Y = \sum_{i=1}^{n} X_i W_i$$

An activation or transfer function combines the inputs coming into a neuron from other neurons and then produces an output based on the transformation function. The relationship between the internal activation level and the output can be linear or nonlinear. One common nonlinear activation function is the sigmoid transfer function that is S-shaped and ranges from 0 to 1. This can be represented in the equation:

$$Y_T = \frac{1}{(1 + e^{-Y})}$$

where Y_T is the transformed (i.e. normalized) value of Y (Sharda, Delen, & Turban, 2014). This is further illustrated in Figure 2.6.

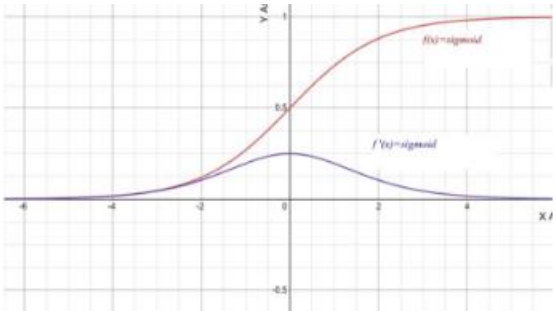


Figure 2.6: Sigmoid activation function (Adapted from Kaloev and Krastev, 2021)

2.3.2. Support Vector Machines

Sharda, Delen, and Turban (2014) define a support vector machine as "a supervised learning method that produces input-output functions from a set of labeled training data". These may be either classification or regression functions. Madhu, et al. (2021) further state that the aim of SVM is to generate a line or a hyperplane that classifies the data and that SVMs are used to improve prediction accuracy and lessen the issue of overfitting. Figure 2.7 shows this structure.

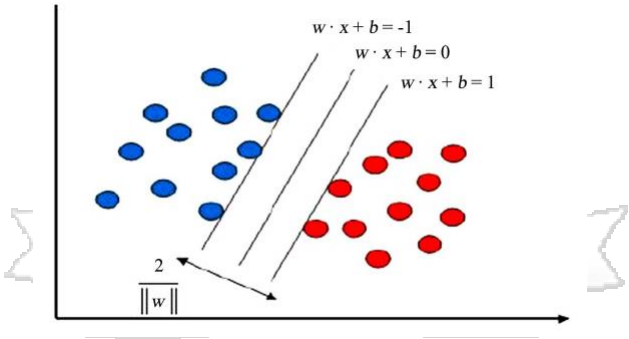


Figure 2.7: Support vector machine margin and hyperplane with trained samples classes (Adapted from Madhu, et al., 2021)

A linear classifier (hyperplane may be represented using the equation:

$$f(x) = w^T + b$$

where X represents the input feature vector of the classifier, w indicates the weight vector, w^T is the transpose of the weight vector, and b holds for the hyperplane position. Here, the margin of the hyperplane is represented by two straight lines and is defined as $m = \frac{2}{\|w\|}$

SVM can find the best compromise between the model complexity and learning ability according to the limited sample information, which minimizes the structural risk and gets the

best learning and generalization ability with the advantages of global minimum point and fast convergence speed. The method can arbitrarily approximate the nonlinear functions, which has great advantages in solving small sample forecasting problems (Yang, He, & Jiang, 2018).

2.3.3. K-Nearest Neighbors (KNN)

KNN predictions are based on the intuitive assumption that objects close in distance are potentially similar. To lend the closest points amongst the k nearest neighbors more say in the outcome of the query point, a set of weights are added to each neighbor (Imandoust & Bolandraftar, 2013). This KNN method is applicable to both classification and regression types of problems. Sharda, Delen, and Turban (2014) postulate that in regression prediction tasks, kNN averages the values of its k nearest neighbors and assigns this result to the case being predicted. This is illustrated in Figure 2.8.

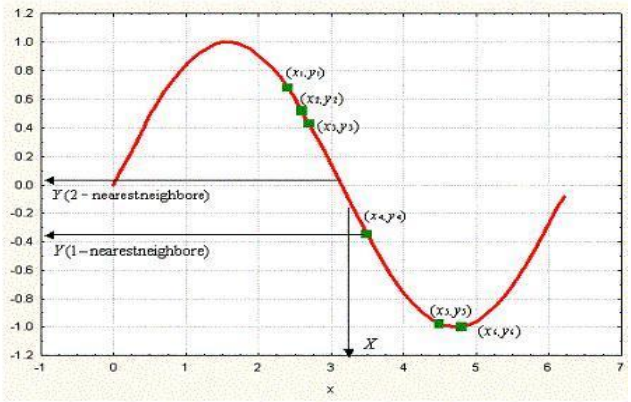


Figure 2.8: The KNN decision rule for regression (Adapted from Imandoust and Bolandraftar, 2013)

A regression kNN problem may be represented by the equation:

$$y = \sum_{i=1}^{k} W(X_0, X_i) y_i$$

where X is the independent variable, Y is the dependent variable and W is a set of weights.

Imandoust and Bolandraftar (2013) give examples of areas where kNN is a suitable data mining technique for performing both classifications and regressions. Notably, they indicate its usage in financial modeling for stock market forecasting to project stock prices based on organizational performance metrics and economic data.

2.3.4. Multiple Linear Regression

The Multiple Linear Regression method models a many-to-one relationship between several independent (input/predictor) variables and one dependent (output/response) variable (Ray, 2019). The MLR model may be represented by the equation:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_m x_m + \varepsilon$$

where y is the dependent variable, x is an independent variable and β_0 is a constant for the regression coefficients (Maulud & Abdulazeez, 2020; Li & Ying, 2011). The least squares method may further be used to estimate the line of best fit between observations in the data (JMP Statistical Discovery, 2023), as shown in Figure 2.9.

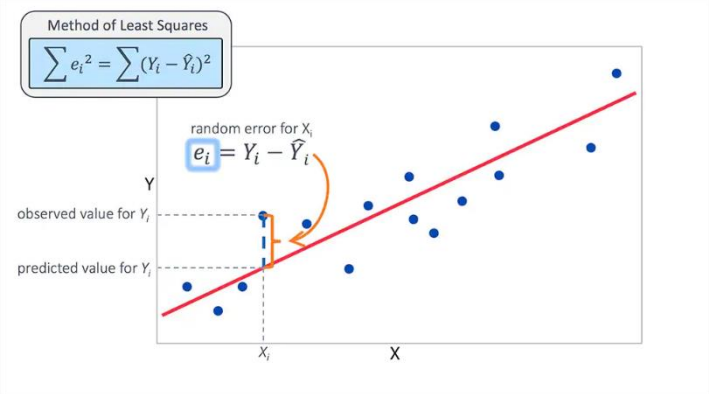


Figure 2.9: Estimation of a regression model using predictor variables (Adapted from JMP Statistical Discovery, 2023)

However, Ray (2019) portends that this method is quite intricate, requires technical know-how on statistical techniques and the sample size used needs to be large enough for one to achieve a high confidence level during analysis.

2.4. Comparison between Traditional and Modern Revenue Forecasting Techniques

Amongst traditional forecasting techniques, there is a clear advantage in using the more mathematical, quantitative approaches over the more subjective kind of forecasting that qualitative techniques offer, since these can be replicated from one scenario to another. Reddick (2004) notes that local government officials rely primarily on judgmental forecasting techniques especially in smaller jurisdictions with a population of 25,000 people or less. On one hand, simple methods require less data, time and expertise from the forecaster. Conversely, complex methods can incorporate a large number of forces and lend themselves to systemic analysis.

ARIMA method gives the highest forecasting accuracy amongst quantitative, non-machine-learning techniques (CEPAL, 2015; Chung, Williams, & Do, 2022; Fattah, Ezzine, Aman, Moussami, & Lachhab, 2018). At the same time ARIMA methods are "data hungry, large sample methods" and the misspecification of this model could result in erroneous predictions because there are so many parameters that need to be accurately evaluated. (CEPAL, 2015). They conclude that since the Exponential Smoothing technique estimates a few parameters, it has a smaller risk of error despite having lower accuracy than a well-specified ARIMA model.

Computer-based forecasting is used to gain high speed and great accuracy in performing forecasting calculations (Onu & Ezeji, 2017). Some commonly used software-based forecasting programs include SIBYL, Stata and IBM SPSS. An advantage of machine learning is that it increases the efficiency of existing services by anticipating demand or targets and adapts services more accurately, enabling more efficient use of resources (Chung, Williams, & Do, 2022).

As far as machine-learning approaches go, each technique has its own merits and demerits based on the type of prediction task, variables considered and level of accuracy expected. In their study, Yang, He, and Jiang (2018) found that a combination of the genetic algorithm (GA) and support vector machine (SVM) models using principal component analysis could wholly incorporate the driving forces of local public finance revenue. Likewise, ANN and SVM ensembles with bagging and boosting outperform a single Neural Network or SVM in its prediction accuracy and this holds true for regression problems (Hajek & Olej, 2010).

On the contrary, ML algorithms are also shown to underperform compared to their simpler, traditional time series forecasting methods. According to Chung, Williams, and Do (2022), one reason is that Machine Learning methods are susceptible to overfitting because they are formulated and trained on the data. Models with low bias in the training data set tend to have higher prediction errors with the test data set. Training and testing data therefore need to have similar data distributions, which may be achieved through preprocessing techniques.

However, Shamsuddin, Sallehuddin, and Yusof (2008) rebound with their observation that an ANN model with few input nodes and their smaller corresponding network structure produces a better forecast result and reduces the chance of over-fitting. A kNN model withstands noisy training data if the training data is large. However, it suffers from computational costs and runtime performance sensitivity to irrelevant or redundant features. This may be avoided by purposeful feature selection or weighting (Imandoust & Bolandraftar, 2013).

2.5. Research Gap

While previous research has focused on different ways to forecast revenue at the local government level, such as the use of existing computer programs like SIBYL (Onu & Ezeji, 2017), or prediction of revenue as a single series, either by singling out one stream of importance (Fikriya & Hikmawati, 2020) or as an aggregated sum of collections over a given period (Yang, He, & Jiang, 2018); there is lack of consensus on the best ML algorithm to use for this problem area, and little mention of taking a composite approach in order to learn the characteristics of individual types of revenue in one jurisdiction, and predicting each in a manner that scales with the variation in sources of revenue as may be seen from county to county.

This study aims to address the gap by applying appropriate machine-learning algorithms to supplement qualitative and quantitative techniques presently used in the context of County Governments in Kenya. This is done by examining past performance and using it as a predictor of future revenue. The results provide a scientific, data-driven approach to mitigate the disparity that exists between the target and actual revenue collected, as evidenced by financial reports (Office of the Controller of Budget, 2022).

2.6. Conceptual Framework

The proposed model was identified after subjecting several ML methods to the same training dataset and running each of them against the test data to determine which one had the highest predictive accuracy for revenue forecasting. Data was obtained from Nairobi City County's automated revenue system and other relevant databases that contained the information required to train the models. There was a preliminary data cleaning and preprocessing stage to scale measurements that may have skewed results, and to reduce noisy datapoints. The input variables were then selected and new features engineered as needed for each respective ML model. Some holdout was done on the data such that approximately 70% was used when training, 20% was used to validate and the remaining 10% was used to test the model.

Regularization was necessitated by performance outcomes, and achieved by tuning model hyperparameters to improve predictive ability, and thereafter, running the train-test-validate cycle once more. At the tail end of the process, the results obtained across various ML models were compared against actual/target values to see which model performed the best and therefore had a good generalization on the revenue dataset. Subsequent monthly forecasts were generated as an output, and recommendations made based on this analysis. Figure 2.10 illustrates the conceptual framework for this approach, as adapted from Madhu, et al. (2021).

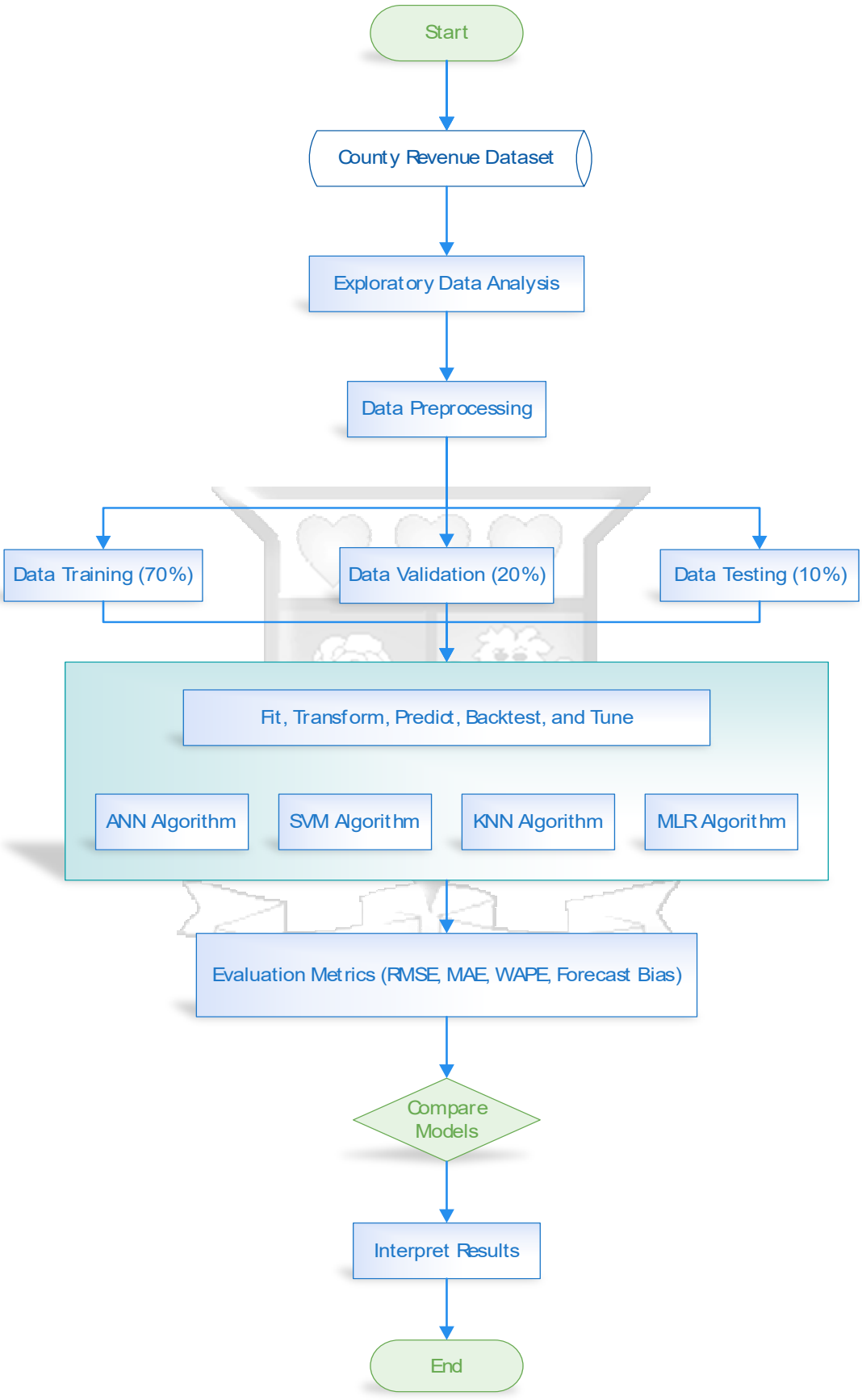


Figure 2.10: Conceptual Model for the Development, Training and Evaluation of a Machine Learning Revenue Forecasting Model

3. Research Methodology

3.1. Introduction

This chapter distills the procedures and gaps identified in the literature review, and sets up the boundaries for running an experiment to validate prevailing theories. The research methodology is further presented in light of the proposed objectives, background study in the forecasting domain, and methods used in similar research. Some key sections under consideration include the research design, where approaches used in this study are identified; system development methodology, where a lifecycle appropriate for the study is identified and research quality, where the criteria for measuring the accuracy of ML models is established.

3.2. Research Design

The research design acts as the blueprint outlining the study's methodology and linking between the proposed objectives and observed results (Dennis, Wixom, & Roth, 2012). This study was conducted through an experimental design approach. This is both a scientific and statistical method wherein the researcher formulates a research question and testable hypothesis, defines dependent and independent variables, collects data from a population sample and subjects these to the experiment's conditions, in order to infer causal relationships that may exist between variables (Bell, 2009).

In this study, several ML models were generated to forecast revenue collection for Kenyan local government authorities, based on historical data from Nairobi City County. Several revenue streams and their influencing factors were analyzed and run through the models to get a prediction of subsequent months' values.

3.3. Population and Sampling

3.3.1. Population

The target population for the study comprised of different revenue streams at Nairobi City County, with records collected over the past 5 years spanning from FY 2018/2019 to FY 2022/2023. The model's data points were sourced from the county's revenue management

system. This county was chosen because of its ease of accessibility, which was advantageous when collecting primary data at source.

3.3.2. Sampling

The study employed non-probability sampling to select the sample. This technique refers to a case where the researcher intentionally chooses items for study "without a basis for estimating the probability that each item in the population is included in the sample" (Kothari, 2004). It is also referred to as purposive sampling and is mainly used to collect focused information. In this study, it was used to analyze own source revenue (OSR) data from NCC, of which there are 11 main categories, as shown in Table 3.1 (Nairobi City County Government, 2022).

Table 3.1: 4th Quarter Revenue and Expenditure Summary for FY 2021/2022 (Adapted from Nairobi City County, 2022)

VOTE R5310000000 NAIROBI CITY COUNTY
4TH QUARTER REVENUE AND EXPENDITURE SUMMARY FOR FY 2021/2022

1	ITEM	Approved Budget	Supplementary 1	Supplementary 2	Quarter 1	Quarter 2	Quarter 3	Quarter 4	Total	% Performance
2	**EXTERNAL SOURCES**									
3	Equitable Share	19,250,000,000	19,250,000,000	19,250,000,000	3,176,196,773	4,812,419,353	1,539,974,193	8,181,112,900	17,709,703,219	0.92
4	WORLD BANK-THS	87,492,017	87,492,017	87,492,017	-	-			-	0.00
5	**DANIDA-UHC**	35,272,875	35,272,875	35,272,875	-	-			-	0.00
6	ASDSPII	36,639,733	36,639,733	36,639,733	-	-			-	0.00
7	UNFPA	7,386,704	7,386,704	7,386,704	-	-			-	0.00
8	**SUB-TOTAL**	**19,416,791,329**	**19,416,791,329**	**19,416,791,329**	**3,176,196,773**	**4,812,419,353**	**1,539,974,193**	**8,181,112,900**	**17,709,703,219**	0.91
9	**OWN SOURCE REVENUE (OSR)**								-	
10	Land Rates	7,458,283,311	7,458,283,311	7,458,283,311	172,219,574	680,982,510	1,400,490,317	228,090,552	2,481,782,954	0.33
11	Parking fees (total)	3,025,000,000	3,025,000,000	3,025,000,000	358,406,316	508,710,028	570,558,481	440,218,332	1,877,893,157	0.62
12	Single Business Permits	2,750,000,000	2,750,000,000	2,750,000,000	106,005,662	319,303,712	809,079,598	133,701,052	1,368,090,024	0.50
13	Plans and Inspections (Building Permits)	1,500,000,000	1,500,000,000	1,500,000,000	141,912,534	174,749,504	142,284,423	133,928,473	592,874,934	0.40
14	Billboards and advertisements	1,200,000,000	1,200,000,000	1,200,000,000	185,545,549	212,635,139	366,143,723	166,688,521	931,012,932	0.78
15	House and Stall Rent	606,000,000	606,000,000	606,000,000	99,272,854	93,785,826	133,714,559	112,350,737	439,123,976	0.72
16	Fire Inspection Certificates	450,000,000	450,000,000	450,000,000	19,653,155	36,934,945	115,415,082	20,994,547	192,997,729	0.43
17	Food Handlers Certificates	250,000,000	250,000,000	250,000,000	31,391,465	30,006,157	49,860,452	23,342,804	134,600,878	0.54
18	Markets	538,770,000	538,770,000	538,770,000	102,864,602	99,934,155	101,169,065	78,342,374	382,310,196	0.71
19	Liquor license	250,000,000	250,000,000	250,000,000	64,753,955	73,697,585	71,805,912	55,607,434	265,864,886	1.06
20	Other Incomes	1,582,691,360	1,582,691,360	1,582,691,360	156,755,294	137,775,195	138,212,269	139,510,455	572,253,212	0.36
21	**TOTAL (OSR)**	**19,610,744,671**	**19,610,744,671**	**19,610,744,671**	**1,438,780,960**	**2,368,514,756**	**3,898,733,881**	**1,532,775,281**	**9,238,804,878**	0.47
22	**TOTAL REVENUE**	**39,027,536,000**	**39,027,536,000**	**39,027,536,000**	**4,614,977,733**	**7,180,934,109**	**5,438,708,074**	**9,713,888,181**	**26,948,508,097**	
23	**Opening Cash balances FY 2021/22**	**600,000,000**	**600,000,000**	**600,000,000**	**299,248,511**				299,248,511	0.50
24	**Sub-total (Cash balances)**	**600,000,000**	**600,000,000**	**600,000,000**	**299,248,511**	**0**	**0**	**0**	**299,248,511**	
25	**Total Cash Available Resources**	**39,627,536,000**	**39,627,536,000**	**39,627,536,000**	**4,914,226,245**	**7,180,934,109**	**5,438,708,074**	**9,713,888,181**	**27,247,756,609**	

The sampling method helped to narrow down factors that contribute to accurate forecasting and inform the modeling process. It was preferred for this study because it gave the researcher some degree of autonomy to select typical and useful cases only, while saving time and money (Oso & Onen, 2009). A key driver for this selection was the fact that the researcher was an individual with limited time resource available to him, and this method offered convenience.

3.3.3. Data Collection

Primary data was used in this research, mainly consisting of revenue data extracted from the county's automated revenue collection and management system. It was expected that the data

will contain billed and actual collections per structured revenue stream, penalties and waivers, county enforcement metrics, rate payer categories and their compliance levels, collection zones and previous year(s) performance percentages against budgeted amounts. However, a limited set of attributes were available, which included monthly collections and number of transactions per revenue stream. The study excluded collecting data on the County's expenditure since this was beyond the scope of the research.

Caution was taken to ensure this data meets the criteria of reliability, suitability and adequacy (Kothari, 2004). In this case, the data tools of choice were document analysis, which according to Oso and Onen (2009) is the thorough assessment of publicly available or privately stored records pertaining to the issue under investigation. This approach has the advantage of convenience when accessing the data sources, while saving time and expenses. Potential selection and coverage biases were mitigated by ensuring that the revenue streams examined were aligned with the county's reporting format at national government level (Office of the Controller of Budget, 2022).

3.4. Data Analysis Methods

Once the data was obtained and collated into an appropriate format, an initial exploratory data analysis was performed to determine its shape, measures of central tendency and expose any outliers. Missing values were treated using various tools available e.g. Pandas library in Python. Feature engineering was employed to extract new values that were deemed appropriate as inputs for different ML algorithms.

The quantitative research data collected underwent inferential analysis after it was processed (Oso & Onen, 2009). Multiple Regression Analysis was used in the study, whereby the dependent variable (monthly revenue collection) was predicted based on its correlation with several independent variables (Kothari, 2004). This was carried out as a multiple independent time series analysis.

3.5. System Development Methodology

System development is typically carried out using an outlined process that guides the activities and stages covered during the entire lifecycle. Methodologies vary in length and breadth

depending on the business scenario, time available and level of expertise of both users and analysts in capturing elements of the field of application. Avison and Fitzgerald (2006) describe the following as attributes that are expected in a methodology:

i) A sequential series of phases/activities stemming from the feasibility study through to acceptance and support.

ii) A costs and benefit analysis of possible solutions and some means to formulate the detailed design necessary to develop computer applications.

iii) A series of software tools to help the system analyst(s) in their work.

iv) A training scheme so that analysts and other actors new to their roles and responsibilities could adopt the newly implemented solution.

v) An implied philosophy that computer systems are relatively good solutions to organizational problems and processing tasks.

Granted, the choice of a particular system development methodology can be a daunting task. There are pros and cons for adopting any to a given type of project implementation. Table 3.2 describes differences in existing approaches (Dennis, Wixom, & Roth, 2012).

Table 3.2: Criteria for selecting a methodology (Adapted from Dennis, Wixom, and Roth, 2012)

	Structured Methodologies			RAD Methodologies		Agile Methodologies
Ability to Develop Systems	Waterfall	Parallel	Phased	Prototyping	Throwaway Prototyping	XP
with Unclear User Requirements	Poor	Poor	Good	Excellent	Excellent	Excellent
with Unfamiliar Technology	Poor	Poor	Good	Poor	Excellent	Poor
that are Complex	Good	Good	Good	Poor	Excellent	Poor
that are Reliable	Good	Good	Good	Poor	Excellent	Good
with a Short Time Schedule	Poor	Good	Excellent	Excellent	Good	Excellent
with Schedule Visibility	Poor	Poor	Excellent	Excellent	Good	Good

The output of this study was not informed by explicitly defined user specifications since it was the product of research and hypotheses on models used in forecasting County Government revenue. The predictor variables were collected from primary sources of data and used to formulate the model within a relatively short project duration. The users who would validate the system during its conception and design were few. There was good awareness of the implementation schedule, which was outlined by the phases through which the model was

developed. These factors therefore lent themselves towards one category of software development methodology, that is, Rapid Application Development (RAD).

3.5.1. Rapid Application Development (Prototyping)

There are two approaches to performing RAD; Prototyping and Throwaway Prototyping. Throwaway prototyping has a detailed analysis phase that aims to gather requirements and generate ideas for the system concept. The purpose of a design prototype is not to come up with a working system. It only contains enough detail to enable users to understand the issues under consideration. Compared with system prototyping, it may take longer to deliver the final system using this approach. Therefore, it was with these considerations in mind that the Prototyping method of system development was adopted. In this method of prototyping, the analysis, design, and implementation stages are all completed at the same time in order to swiftly provide a condensed version of the proposed solution and present it to the users for review and comment (Dennis, Wixom, & Roth, 2012). This flow is shown in Figure 3.1.

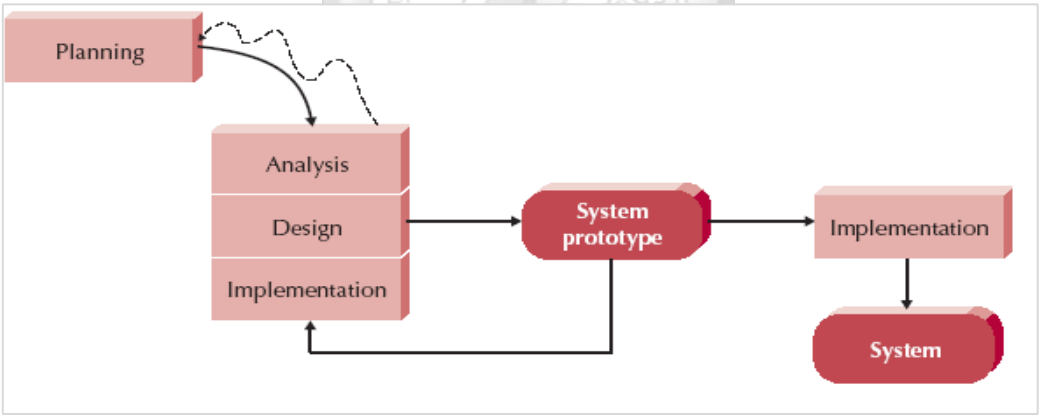


Figure 3.1: System Prototyping Methodology (Adapted from Dennis, Wixom, and Roth, 2012)

Prototyping can also be used to gather requirements about the system under review. This typically cuts in half the time that requirement determination normally takes, with better results (Avison & Fitzgerald, 2006). Time was a critical factor for this research since it was not possible to go through the entire formal process of system design and implementation, as is commonplace in methodologies like the waterfall approach. Creation of a prototype becomes a quick and dirty way of coming up with basic functionality that can be tested with users whose comments are reintegrated in the next iteration of the system until a final version is arrived at (Dennis, Wixom, & Roth, 2012).

3.5.2. System Analysis

Avison and Fitzgerald (2006) describe analysis as “an attempt to understand all aspects of the present system, why it developed as it did, and eventually indicate how things might be improved by any new system”. In the study, this involved determining features that users needed to be implemented in the final system. It was assumed that the county under review had already rolled out an automated revenue management system that constantly records actual revenue collected per revenue stream. The stored historical data formed the basis for future predictions using multiple regression analysis and machine learning algorithms, which could then augment existing system functionality.

3.5.3. System Design

The design phase establishes the hardware, software, and network infrastructure that will be used, as well as the user interface, forms, and reports that will be implemented as part of the system's functionality. It also identifies the particular modules, databases and files required (Dennis, Wixom, & Roth, 2012). It was intended that this section of Prototyping would borrow heavily from the analysis performed in the preceding stage, where users described what requirements should go into the system. An appropriate data model was constructed that would contain the attributes needed for the forecasting process. The formatting was specified and sample interfaces designed. No hardware components were envisioned as part of the solution other than the base computing platform on which the models would be run (which may have been local or cloud-based).

3.6. System Implementation

3.6.1. Model

Seven predetermined models were built side by side and trained on approximately 70% of the data obtained from the county. The application accepts input either from flat file storage (CSV-delimited) or from a relational database and processes this data through scripts for further analysis and prediction. There was no reliance on APIs to load raw data from existing revenue management systems into the system in real-time owing to the need to first process and transform the data as per expected input variables (features).

3.6.2. Test Bed

A test bed was set up with about 10% of the dataset reserved for this exercise. It was run locally using the trained ML models and variables processed from the county revenue data. This was done in a Python-based environment. Prior to this, model validation was performed using about 20% of the dataset.

3.6.3. Programming Language

The coding language of choice was Python. It has robust ML libraries and extensive support for third party visualization tools which may be required to present the results of the prediction process. It also has in-built statistical tools that give a fair representation of the models' accuracy and performance. A front-end tool was also built to help users interact with the models via a web-based GUI.

3.7. Research Quality

StudySmarter (2023) defines quantitative research quality criterion as evidenced by internal and external validity, reliability, and objectivity. These three pillars were adhered to so that potential biases were eliminated and the results of the study could be replicated by other researchers.

Statistical evaluation metrics were used to gauge the accuracy of machine learning algorithms used in the study and the predicted values' confidence interval. Back testing with cross-validation was used to calculate an average error metric, and hence the summary score for the regression method. Amongst all these ML models, hyperparameter tuning was done to improve the predictive power of each algorithm before their results could be compared (Raschka, 2018).

3.8. Risk-Benefit Analysis

This research was deemed to be of low risk since no human or animal biological trials were involved. Rather, data from computerized systems was collected, evaluated and run through machine learning models. The outcome of this research could be beneficial to the local government administration in Nairobi, who may gain new insights on potential quantities of revenue that could be unlocked in the city. These methods may be replicated across Kenya by

other county governments to enhance their forecasting capabilities and thereby assist them to meet targets and be more self-sufficient. Table 3.3 shows the current 3-year OSR forecasts at NCC from FY 2023/24 to FY 2025/26 (Nairobi City County Government, 2023).

Table 3.3: Fiscal Framework for FY 2023/2024 & Medium Term (Adapted from Nairobi City County, 2023)

ITEM			
	FY 2023/2024	FY 2024/25	FY 2025/26
REVENUES			
EQUITABLE SHARE	19,782,840,133	20,811,547,820	21,893,748,307
CONDITIONAL GRANTS			
Own Source Revenues	19,990,072,415	21,060,926,033	22,911,722,335
Unutilized CRF balances from FY 2021/2022	994,291,212		
Total Revenues + cash balances	**40,767,203,760**	**41,872,473,853**	**44,805,470,642**

3.9. Dissemination and Utilization of Results

The results of this research will primarily be accessible via Strathmore University's research and publications *DSpace* portal, where it may be retrieved by institutional members or the general public if they have sufficient rights granted to them. The results will also be shared with Nairobi City County for their analysis and consideration in future studies and/or decision-making processes.

3.10. Ethical Considerations

The research utilized as much data from public record as possible; mapping input variables from government reports on annual revenue collected at the county level. Where selected revenue streams' data was found to be poorly documented or input variables insufficient for modeling, ethical approval was requested to collect primary data from the relevant county finance departments for use in this study. Confidentiality was maintained where data provided was deemed sensitive to ongoing county development initiatives, or where disclosure may have borne consequence to the County's obligations to its citizens and/or exposed personally identifiable information.

4. System Analysis and Design

4.1. Introduction

System analysis and design is a methodical process for developing high-quality information systems. Analysis aims to develop a logical model of a new system while design purposes to construct a physical model that satisfies all of the system's needs (Tilley & Rosenblatt, 2016). This section contains details about the proposed model's design and overall architecture. It describes the requirements that the system should address, the various components considered when designing the model and how each interacts to achieve the desired objectives. The standard representation used is the Unified Modeling Language (UML v2.5.1). The items that achieve this design representation include a use case diagram, class diagram, activity diagram, sequence diagram, entity-relationship diagram, data flow diagram and wireframes.

4.2. Requirements Analysis

Requirements gathering and analysis involves the elicitation of needs for a system from stakeholders by observing their organization's business processes, identifying gaps for which a software-based solution may be suitable, documenting features of the solution, also known as requirements, while weeding out anomalies, inconsistencies and incompleteness (Mall, 2018). This requirements analysis highlights the features that should be addressed in the application in line with the research objectives. These features are covered in two categories: functional and non-functional requirements.

4.2.1. Functional Requirements

Functional requirements describe processes the system has to perform or information it needs to contain (Dennis, Wixom, & Roth, 2012). The following are the functional requirements of the proposed solution:

i) The system should have well validated input formats for data used in running forecasts.

ii) The system should present the user with the available ML model parameters to use when analyzing the data input.

iii) The system should present the projected revenue figures for the subsequent fiscal periods.

4.2.2. Non-Functional Requirements

Non-functional requirements describe behavioral properties that the system must have (Dennis, Wixom, & Roth, 2012). The following are the non-functional requirements of the proposed solution:

4.2.2.1. Performance

Depending on the type of machine learning model being run, there can be great demand placed on the CPU of the operating environment. The application should have a reasonable response time when run on a 64-bit multi-core processor. Normalization and standardization may also be used to scale features and help reduce training time through fewer iterations and faster convergence.

4.2.2.2. Accuracy

The application should check uniformity across the input data depending on the type of model used. Where units of measurement vary, preprocessing by scaling should be done on numeric fields so that they fall between 0 and 1, and encoding for categorical variables to give them numeric representations. This will ensure that consistent results are obtained whenever predictions are made from any given dataset.

4.2.2.3. Usability

The application should be user friendly, validate inputs given via CSV or XSLX and it should present its results in a consistent format. The models tested should be well documented.

4.3. System Architecture

The system comprises of a web application that has three layers, i.e. a front-end user interface (UI), some middleware with the application logic and the back-end framework. The user interacts with the system through a web browser where they input their own source revenue stream data for which a prediction is required. This may either be done via a form on the UI or by uploading a file that has the required fields specified in a template. The WSGI-based (Web Server Gateway Interface) Python application then ingests the data, generates features and passes them onto a machine learning model for inference. The prediction results obtained are then stored in a database and visualized in form of a graphical forecast depending on the projected period needed. This workflow is depicted in Figure 4.1.

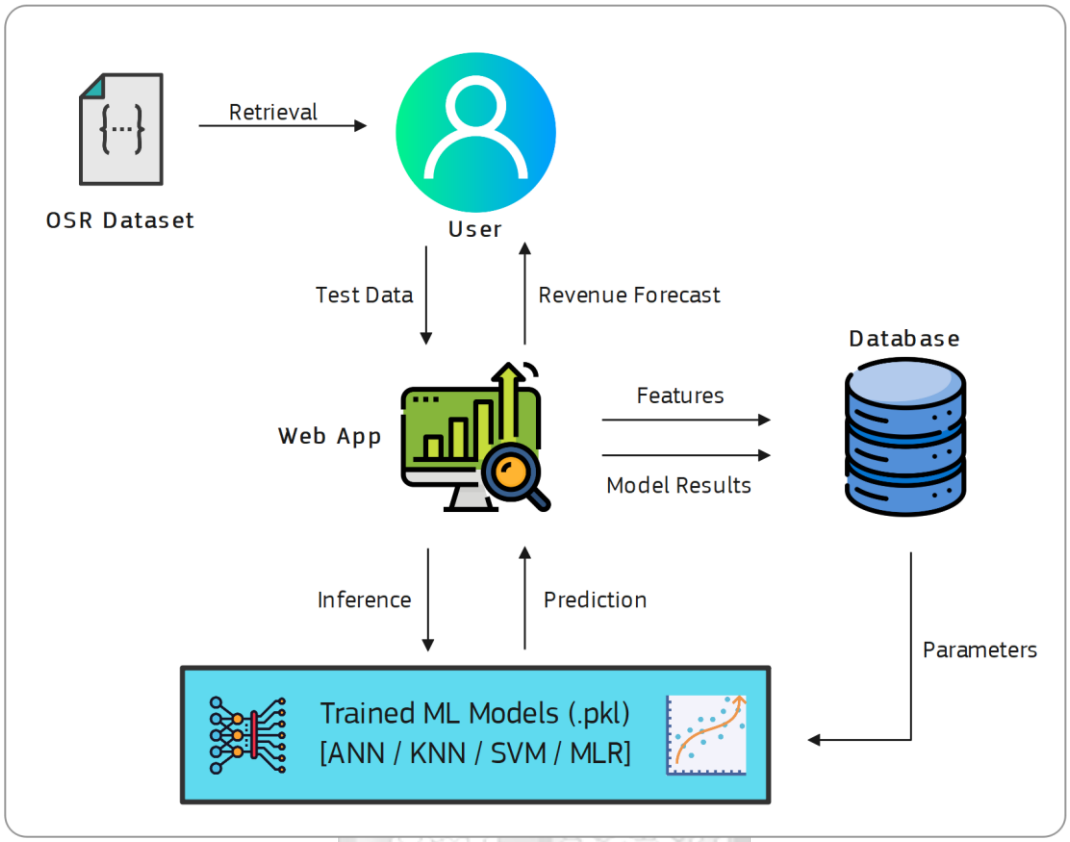


Figure 4.1: System Architecture

4.4. System Design

The design describes how the application will achieve its requirements in more detail. It captures who the system users are, what inputs will be submitted to the system, how the different system components will interact with each other and how the data is manipulated during execution to deliver the desired output. The respective design representations are demonstrated in the following sub-sections.

4.4.1. Use Case Diagram

The use case diagram represents the main actors of the system and the functions they are able to execute through features inherent in the application. The system reacts to a request made by a principal actor that is related to a goal (George & Valacich, 2016). The two actors identified are the system user and administrator. The system use case diagram is shown in Figure 4.2.

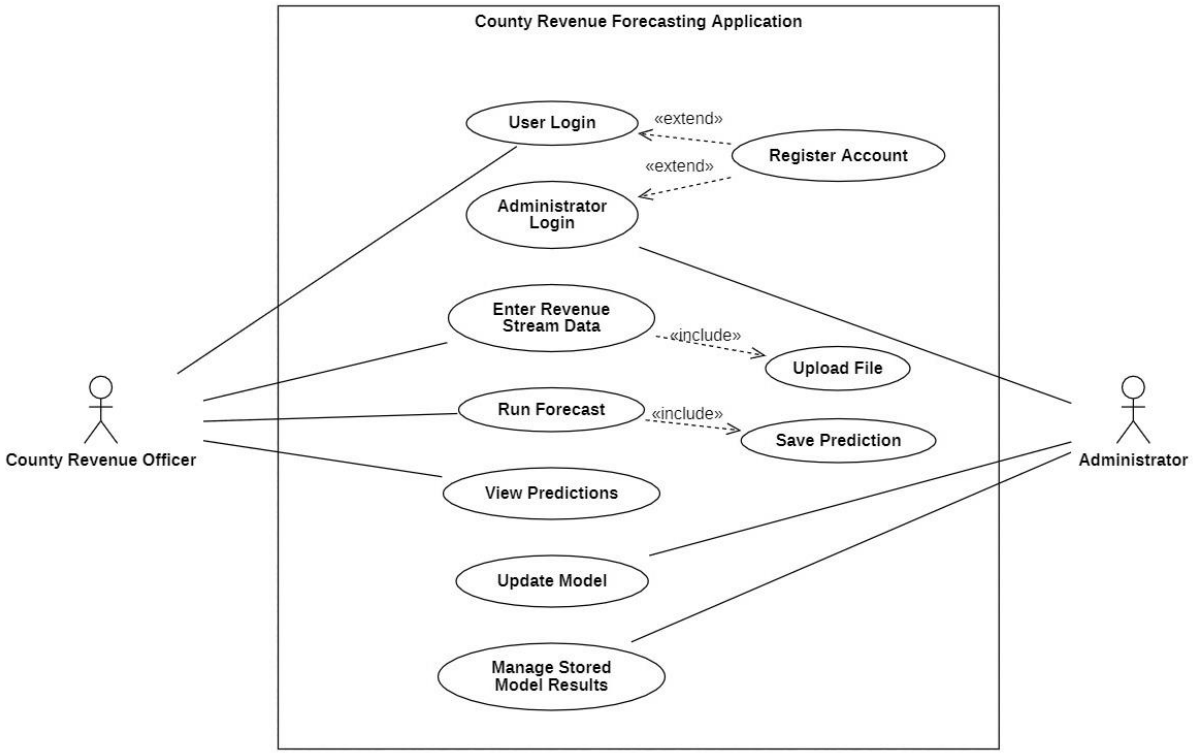


Figure 4.2: Use Case Diagram

Detailed use case descriptions of system functions are further provided in Tables 4.1 to 4.7.

Table 4.1: User Login Use Case Description

Use Case Title: User Login
Primary Actor: County Revenue Officer
Level: Subfunction
Stakeholders: User
Precondition: User accesses the revenue forecasting application
Minimal Guarantee: Rollback of any incomplete transaction
Success Guarantees: User's credentials are authenticated and they are redirected to the home screen
Trigger: User accesses the revenue forecasting application URL
Main Success Scenario: 1. User fills in their username and password on the login form 2. System verifies the details provided
Extensions:

1a. Login page is not available 1a1. User quits application 2a. Credentials provided are invalid 2a1. User quits application 2a2. User resets their password

Table 4.2: Administrator Login Use Case Description

Use Case Title: Administrator Login
Primary Actor: Administrator
Level: Subfunction
Stakeholders: Administrator
Precondition: Administrator accesses the revenue forecasting application
Minimal Guarantee: Rollback of any incomplete transaction
Success Guarantees: Administrator's credentials are authenticated and they are redirected to the home screen
Trigger: Administrator accesses the revenue forecasting application URL
Main Success Scenario: 1. Administrator fills in their username and password on the login form 2. System verifies the details provided
Extensions: 1a. Login page is not available 1a1. Administrator quits application 2a. Credentials provided are invalid 2a1. Administrator quits application 2a2. Administrator resets their password

Table 4.3: Data Input Use Case Description

Use Case Title: Enter Revenue Stream Data
Primary Actor: County Revenue Officer
Level: User Goal
Stakeholders: Revenue Officer

Precondition: User successfully logs in to the revenue forecasting application
Minimal Guarantee: Rollback of any incomplete transaction
Success Guarantees: Data points are captured and stored in the database
Trigger: User accesses the web application home page
Main Success Scenario: 1. User uploads the revenue data set as csv/xlsx format or manually inputs the data via a form on the application interface 2. System validates the data provided 3. System stores the data and generated features in their respective tables on the database
Extensions: 1a. File format used is not supported 1a1. System prompts user to provide acceptable file formats 2a. Data provided contains invalid variable types 2a1. System prompts user to enter correct values 3a. System is unable to store the data in the database 3a1. Transaction rolled back. User tries again. 3a2. Transaction rolled back. User quits application.

Table 4.4: Run Forecast Use Case Description

Use Case Title: Run Forecast
Primary Actor: County Revenue Officer
Level: Summary
Stakeholders: Revenue Officer
Precondition: User submits a revenue stream's data points on the application
Minimal Guarantee: Rollback of any incomplete transaction
Success Guarantees: An inference is made against trained models and a prediction result is returned
Trigger: User submits their revenue stream data attributes
Main Success Scenario: 1. System loads the test data features 2. System loads the specified ML model

3. System runs a test of these features against the chosen algorithm 4. System scores and returns the model results 5. Predicted values and model results are stored in the database
Extensions: 1a. System is unable to read the data in the database 1a1. User gets an error message. Transaction rolled back. User tries again. 1a2. User gets an error message. Transaction rolled back. User quits application. 2a. System is unable to load the specified ML model 2a1. User gets an error message. Transaction rolled back. User tries again. 2a2. User gets an error message. Transaction rolled back. User quits application. 3a. System is unable to run a test of these features against the chosen algorithm 2a1. User gets an error message. Transaction rolled back. User tries again. 2a2. User gets an error message. Transaction rolled back. User quits application. 4a. System is unable to score and return the model results 2a1. User gets an error message. Transaction rolled back. User tries again. 2a2. User gets an error message. Transaction rolled back. User quits application. 5a. System is unable to store the prediction results in the database 3a1. Transaction rolled back. User tries again. 3a2. Transaction rolled back. User quits application.

Table 4.5: View Predictions Use Case Description

Use Case Title: View Predictions
Primary Actor: County Revenue Officer
Level: User Goal
Stakeholders: Revenue Officer, County Executive
Precondition: User runs a forecast with an unseen sample of data
Minimal Guarantee: Rollback of any incomplete transaction
Success Guarantees: Application displays the model results and predicted forecast values
Trigger: User runs a forecast on submitted revenue data
Main Success Scenario: 1. System loads both tabular and graphic visualizations of the predicted results 2. System loads the model results and its measures of predictive accuracy

Extensions: 1a. System is unable to load the revenue predictions on the UI 1a1. User gets an error message. Transaction rolled back. User tries again. 1a2. User gets an error message. Transaction rolled back. User quits application. 2a. System is unable to load the ML model results 2a1. User gets an error message. Transaction rolled back. User tries again. 2a2. User gets an error message. Transaction rolled back. User quits application.

Table 4.6: Update Model Use Case Description

Use Case Title: Update Model
Primary Actor: Administrator
Level: Summary
Stakeholders: Administrator
Precondition: Administrator successfully logs in to the revenue forecasting application
Minimal Guarantee: Rollback of any incomplete transaction
Success Guarantees: Trained models are updated on the application
Trigger: Administrator accesses the web application home page
Main Success Scenario: 1. Administrator navigates to the admin page 2. Administrator redeploys the retrained models
Extensions: 1a. Admin page is not available 1a1. Administrator quits application 2a. System is unable to redeploy the retrained models 2a1. Administrator gets an error message. Transaction rolled back. Administrator tries again. 2a2. Administrator gets an error message. Transaction rolled back. Administrator quits application.

Table 4.7: Manage Stored Model Results Use Case Description

Use Case Title: Manage Stored Model Results
Primary Actor: Administrator
Level: User Goal
Stakeholders: Administrator
Precondition: Administrator successfully logs in to the revenue forecasting application
Minimal Guarantee: Rollback of any incomplete transaction
Success Guarantees: Model results are read or deleted from the database as desired
Trigger: Administrator accesses the web application home page
Main Success Scenario: 1. Administrator navigates to the admin page 2. Administrator reads model results generated in the past 3. Administrator deletes historical results
Extensions: 1a. Admin page is not available 1a1. Administrator quits application 2a. System is unable to load previous model results from the database 2a1. Administrator gets an error message. Transaction rolled back. Administrator tries again. 2a2. Administrator gets an error message. Transaction rolled back. Administrator quits application. 3a. System is unable to delete previous model results stored on the database 2a1. Administrator gets an error message. Transaction rolled back. Administrator tries again. 2a2. Administrator gets an error message. Transaction rolled back. Administrator quits application.

4.4.2. Class Diagram

A class diagram shows the object classes and relationships involved in a use case (Tilley & Rosenblatt, 2016). It describes the member attributes under consideration when designing the system. Classes represent the main concepts at play within the predictive model, highlight the level of inheritance and show relationships and multiplicity between objects. These are captured in Figure 4.3.

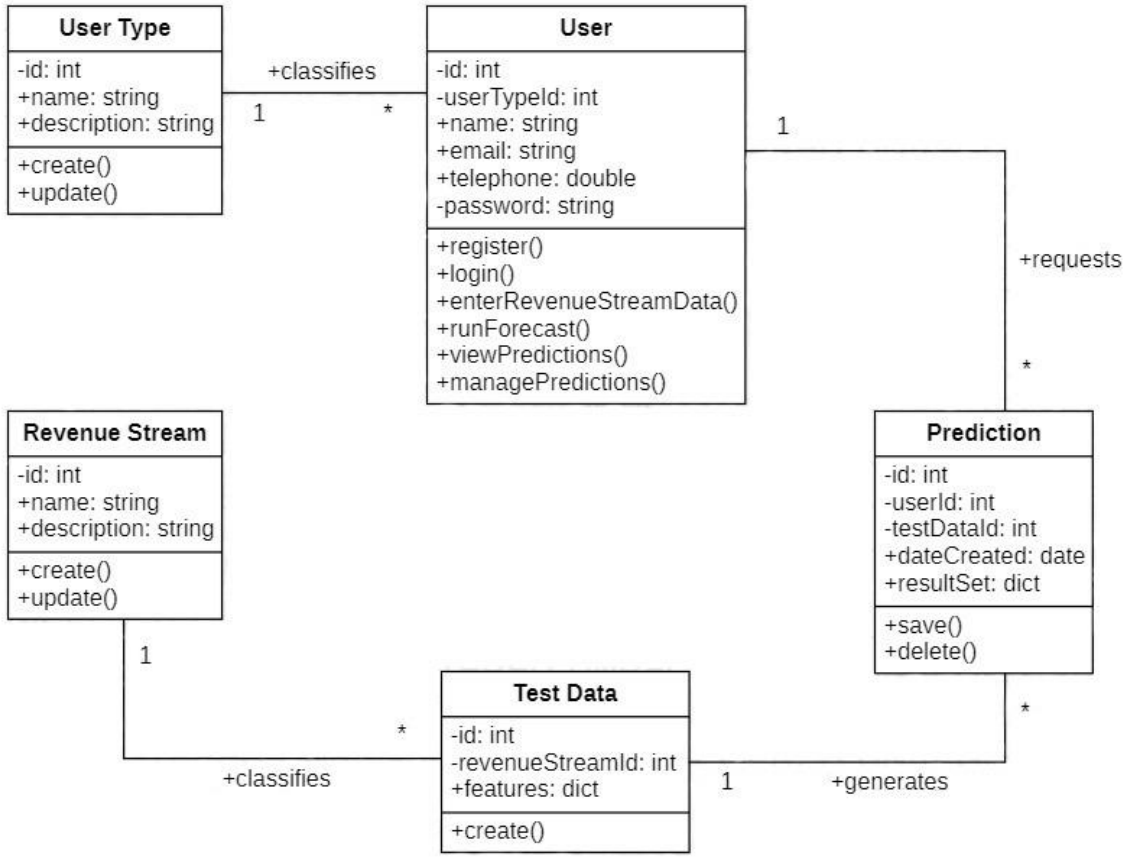


Figure 4.3: Class Diagram

4.4.3. Activity Diagram

An activity diagram shows the conditional logic for the sequence of system activities needed to accomplish a business process (George & Valacich, 2016). An activity is a state represented by a node with a singular action flowing into and one or more transitions emanating from it (Mall, 2018). Figure 4.4 depicts the main execution path through the web application.

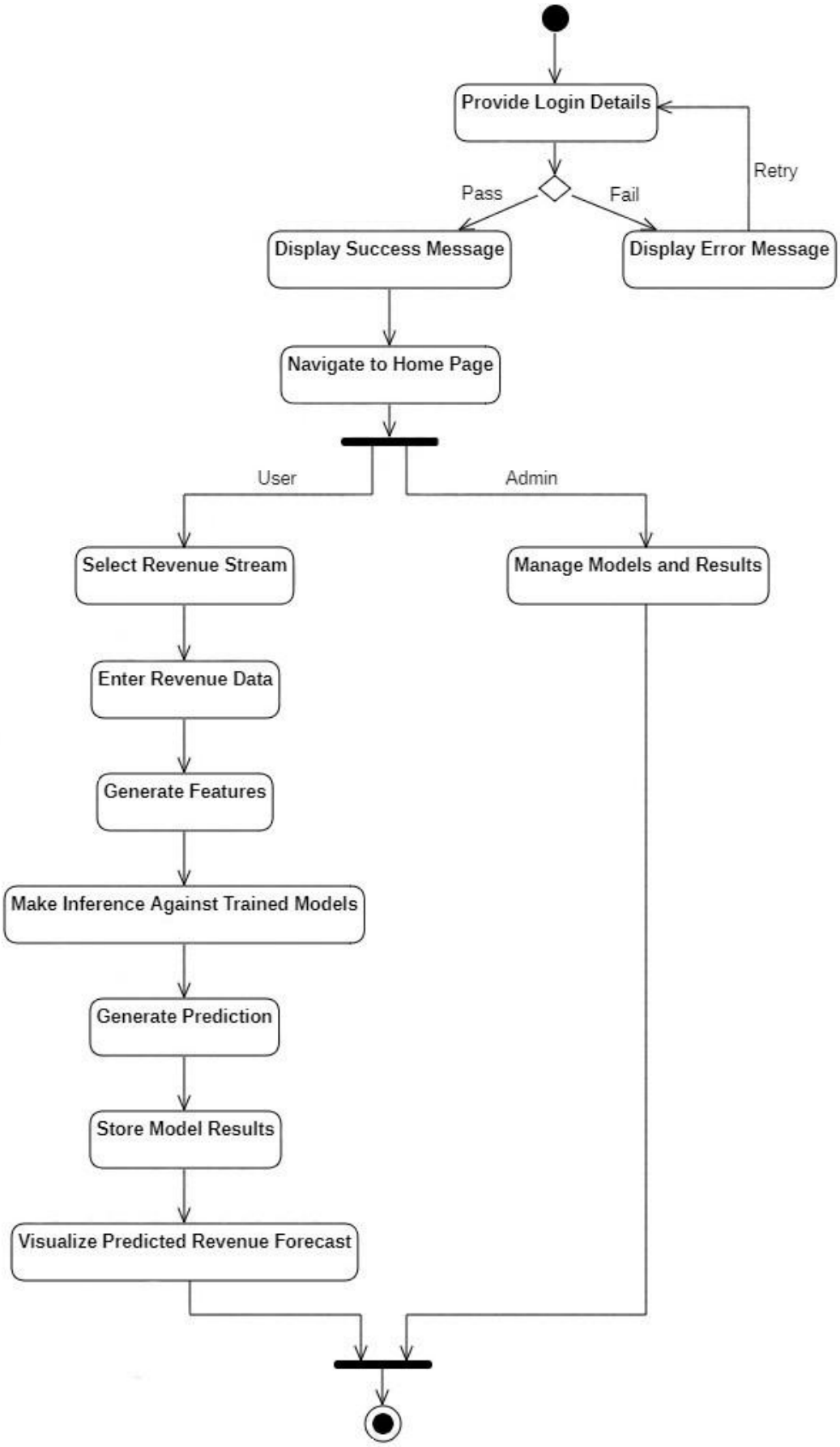


Figure 4.4: Activity Diagram

4.4.4. Sequence Diagram

A sequence diagram depicts the way messages are passed, received and acted upon by the different modules. They can demonstrate how a use case's events or activities map to the operations of object classes in the class diagram. (Lu & Kim, 2011). Four main entities are identified in the application and interactions between them over time are shown in Figure 4.5.

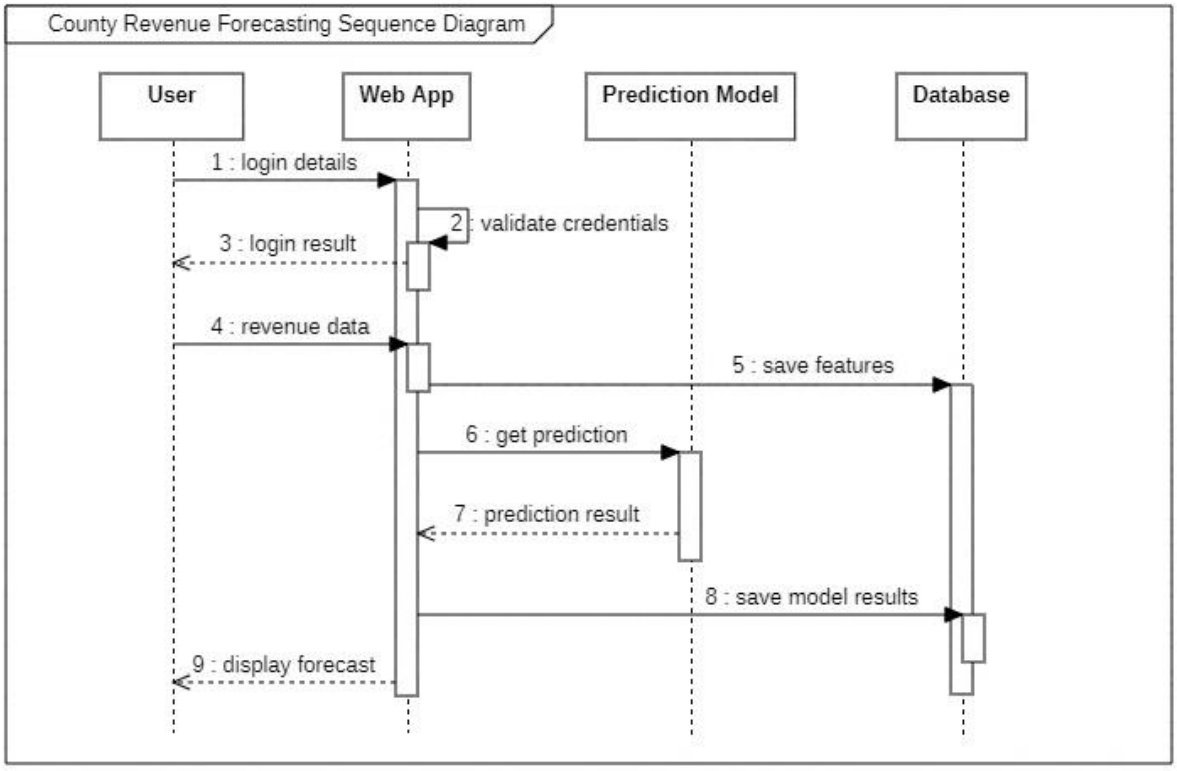


Figure 4.5: Sequence Diagram

4.4.5. Entity Relationship Diagram

An Entity Relationship Diagram (ERD) models the logical structure of a database by showing the associations that exist among tables and their attributes (Delhi University, 2023). Likewise, an Entity-Relationship Model is a comprehensive, logical representation of the entities, associations, as well as data elements for an institution or department (George & Valacich, 2016). Figure 4.6 illustrates an ERD for the prototype revenue forecasting application with five main tables that will exist in the database and different relationships that will be created between them via foreign keys. The goal is to develop a normalized table structure that captures

data attributes at an atomic level, while enforcing functional dependency between these attributes and their respective identifiers.

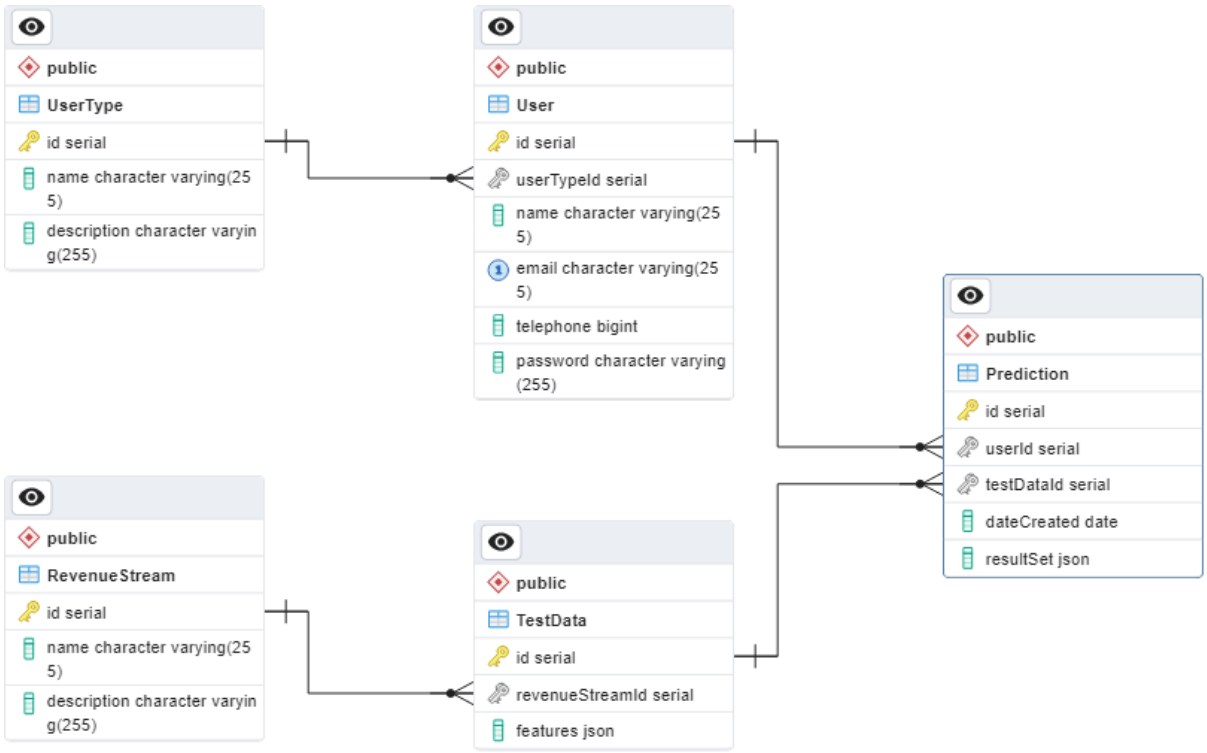


Figure 4.6: Entity Relationship Diagram

4.4.6. Data Flow Diagram

According to George and Valacich (2016), a Data Flow Diagram (DFD) is an illustration of the exchange of data between external entities, and internal processes and repositories found in a system. It provides an hierarchical, abstracted view of the system and can be used to explain the overall system design to others (Aleryani, 2016) without showing the sequence of execution, control flow or actual algorithms used (Mall, 2018).

The revenue forecasting application comprises of two external entities i.e. the Revenue Officer and System Administrator. It has five main processes, one model database as the data store and numerous data flows linking these components. The processes capture the high-level system functions which include: input of test data, generation of features to be passed onto the trained models, the prediction of forecasts and their visualization, and finally procedures for updating the inference models and managing the stored model results. Figure 4.7 depicts the DFD for the built system.

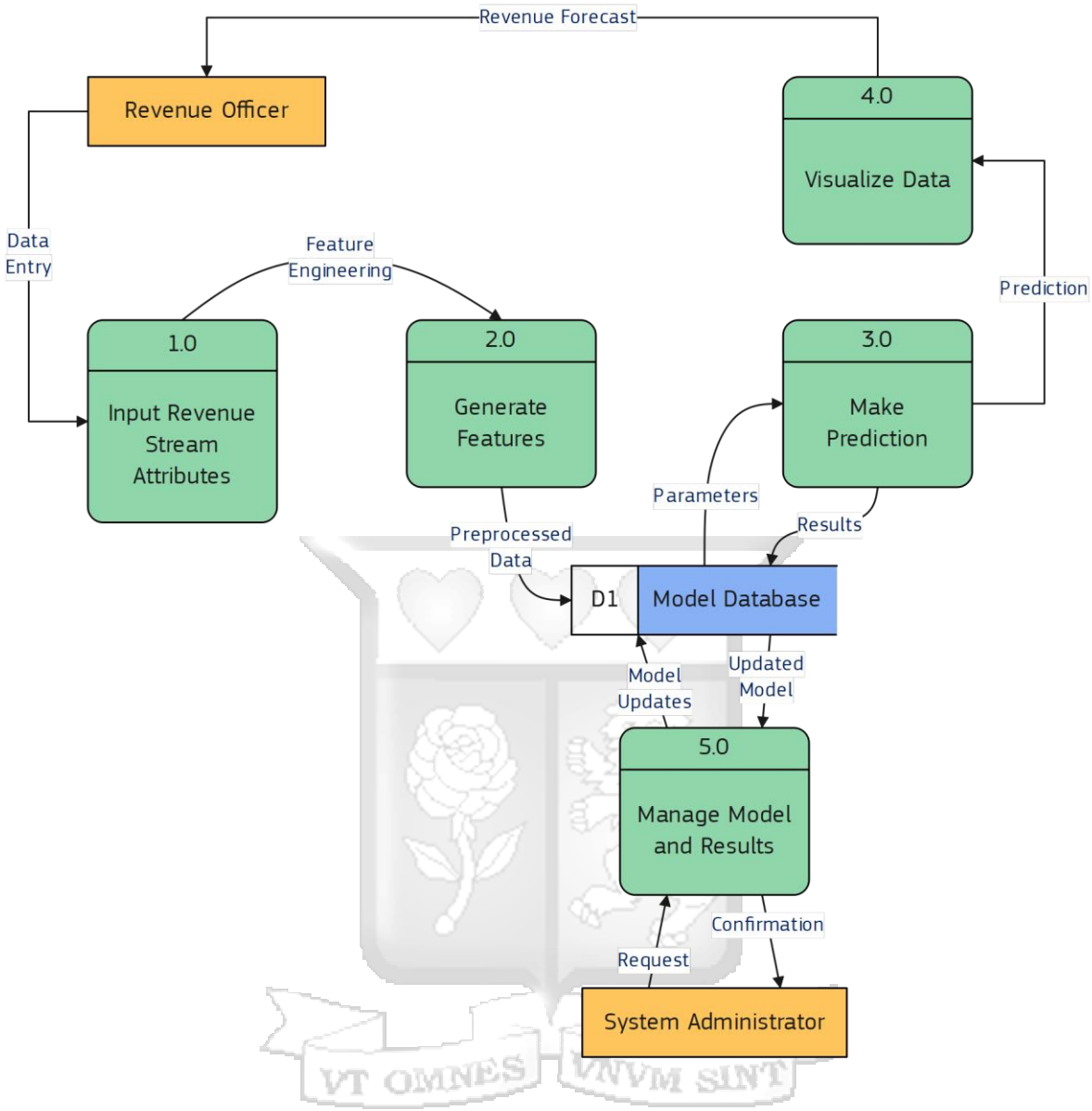


Figure 4.7: Data Flow Diagram

4.4.7. Wireframes

Wireframes are mockup sketches of how forms and screens will look like in the developed system. They help others get a better sense of the visual interface design, flow and navigation through the application. According to Bruton (2022), they also help stakeholders to agree on concepts before the interface is built out with code, while keeping users at the center of focus.

4.4.7.1. Landing Site Wireframe

Figure 4.8 shows the landing site for the web application which users first see when they enter the URL in their browsers.

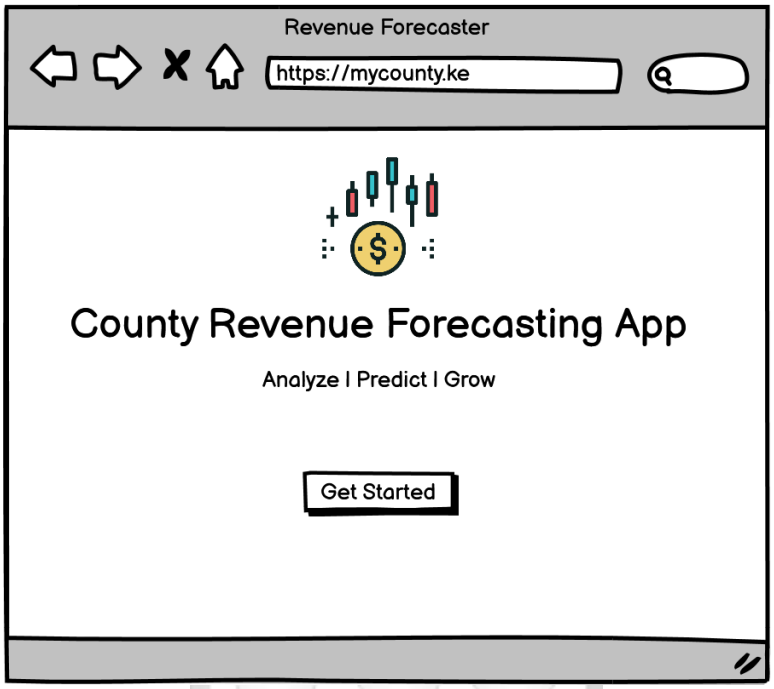


Figure 4.8: Landing Site Wireframe

4.4.7.2. Login Page Wireframe

Users are requested to log into the system in order to access its functionality. Figure 4.9 shows how they are authenticated through a third-party identity service, Sign in with Google. A Google account shall be required, whereby the user's email address will be their username and their password will be used to complete the sign-in process.

Figure 4.9: Login Page Wireframe

4.4.7.3. Home Page Wireframe

The wireframe in Figure 4.10 demonstrates the home page seen after successful log in. It contains links for navigation and narrated steps which may be followed when attempting to generate a revenue forecast.

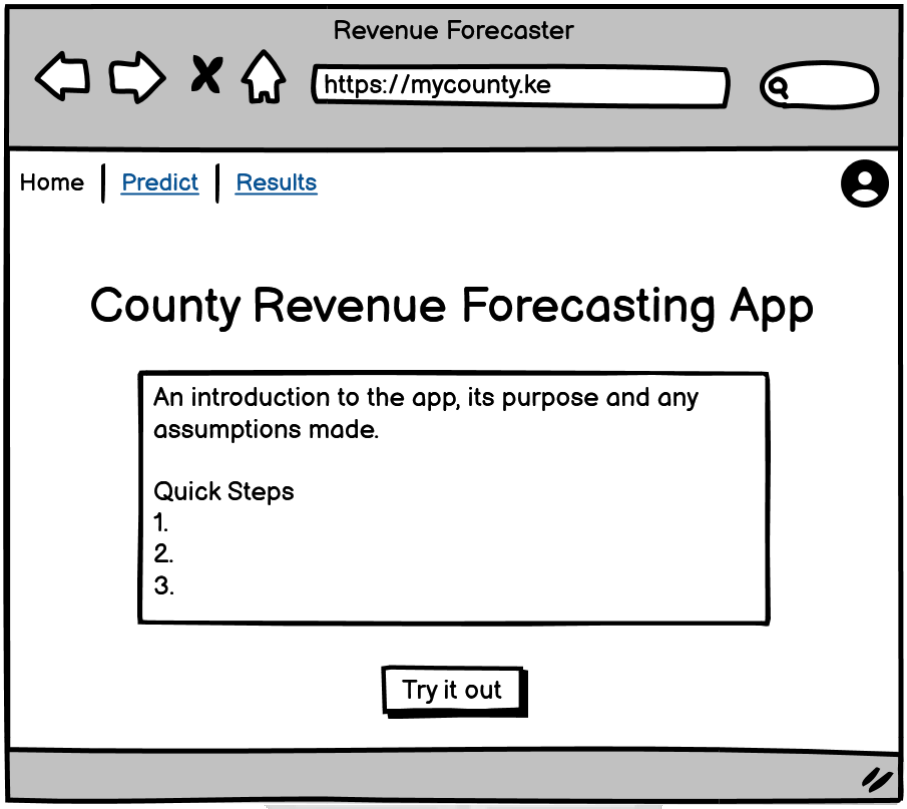


Figure 4.10: Home Page Wireframe

4.4.7.4. Data Entry Wireframe

Figure 4.11 displays a form that enables test data input. To ensure the correct entry of attributes, a user first selects the revenue stream and forecasted period for which they wish to make a prediction. Thereafter, they may choose to either type in the filtered values manually or upload a spreadsheet containing these fields in case the data points are too many. Once done, they may click on the “Run Forecast” button.

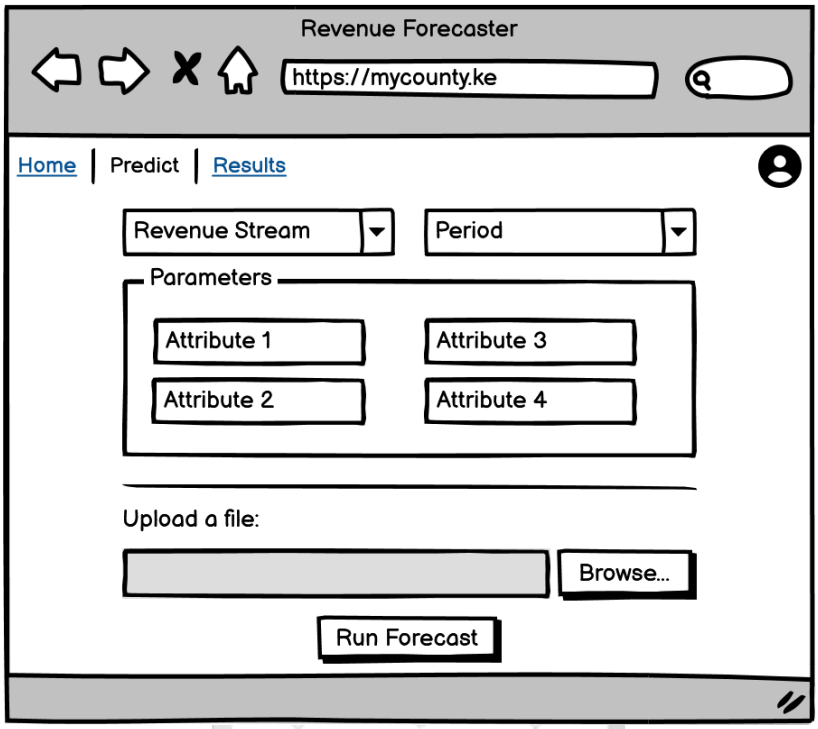


Figure 4.11: Data Entry Wireframe

4.4.7.5. Data Visualization Wireframe

This wireframe contains the visual presentation of prediction results after test data has been fitted and scored against the model. Some sample graphs and tables are shown in Figure 4.12

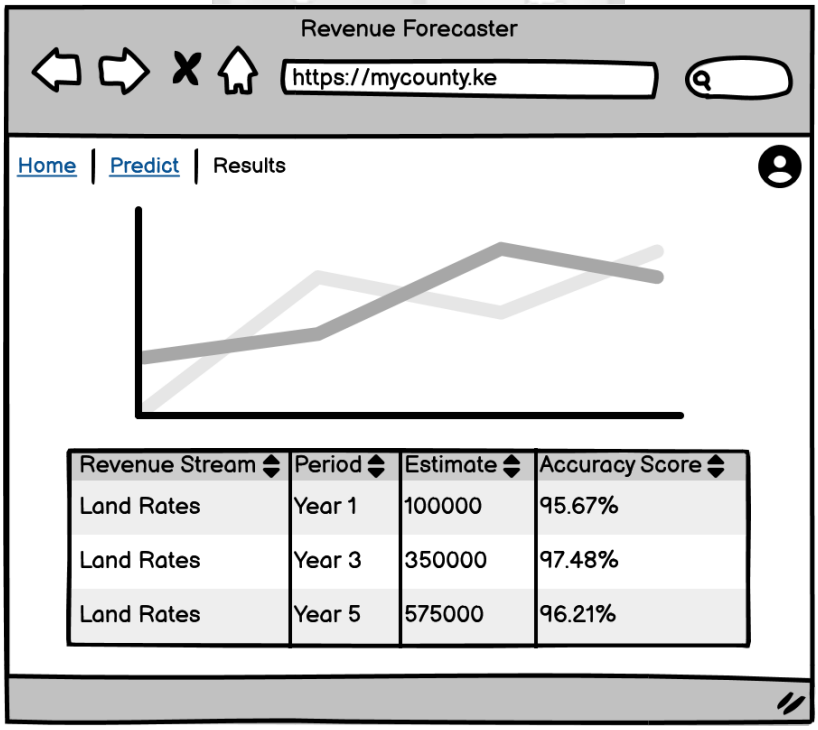


Figure 4.12: Data Visualization Wireframe

5. System Implementation and Testing

5.1. Introduction

The model building process was framed as a Time Series Analysis (TSA) problem whereby a regularly spaced and time ordered historical data set is prepared and used for training machine learning algorithms, that could then predict values over a given forecasting horizon. A comparison was done between statistical and machine learning models to see how they performed on the same data set.

5.2. Model Components

The models that were analyzed were built using forecaster objects from the *skforecast* library (Rodrigo & Ortiz, 2024). These offered a standardized means of encapsulating the features needed to do time series forecasting by passing parameters in a way that can be replicated across different types of algorithms. Figure 5.1 shows the forecaster object components and its methods.

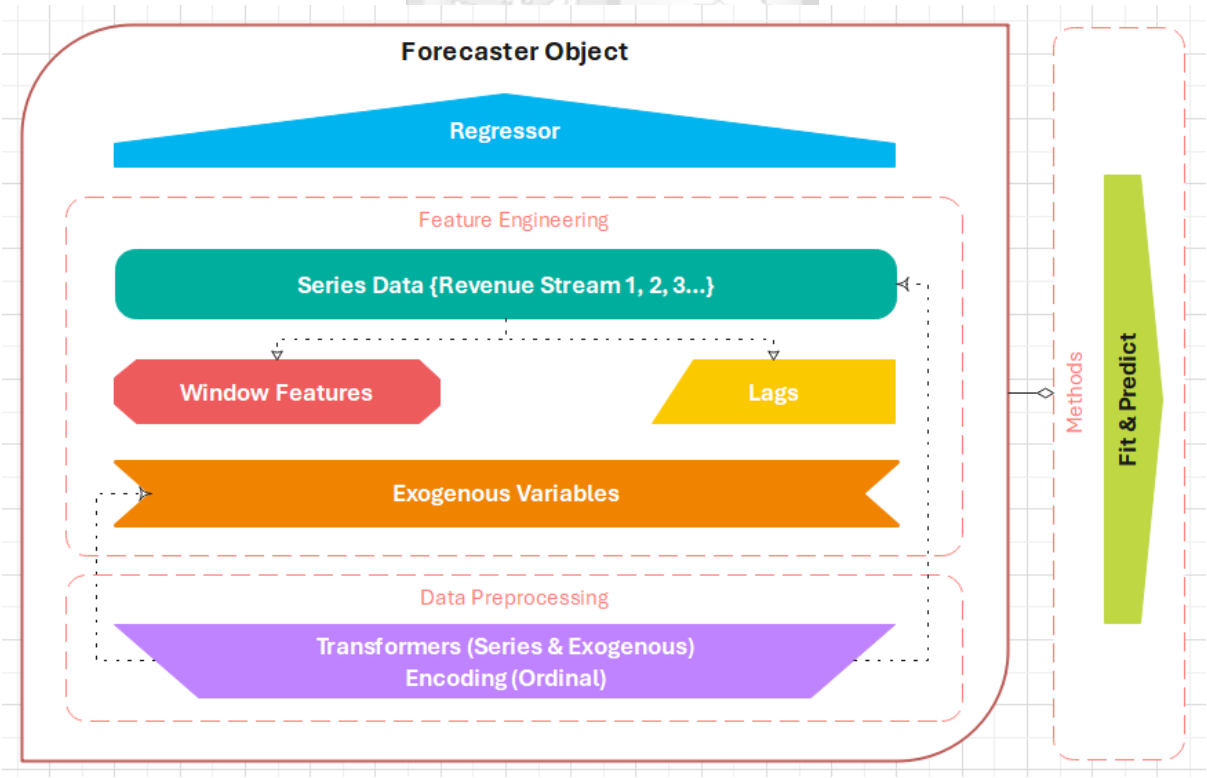


Figure 5.1: A Block Diagram of the Forecasting Model Components and its Methods

5.2.1. Forecaster Components

A forecaster object has different parts used to set the context for which predictions were made. These are attributes found in and derived from the data set itself.

5.2.1.1. Regressor

This is the type of model passed to the forecaster object, fitted to the training data and used to make predictions. The model selection was guided by literature on the topic of study and no assumptions were made as to which would fare best when generating forecasts. Each model was subjected to a similar process of evaluation and compared by their individual performance.

5.2.1.2. Series / Endogenous Variables

These are the dependent variables for which a prediction is made. In this case, they were several types of revenue streams making up the total county revenue. They had to be predicted individually (local models) but approached as a whole (global model). The data set contained thirteen named revenue types and one arbitrary class called "Other Income". This instance helped aggregate collections that were incomplete across the five-year period under consideration. Accordingly, the domain was framed as an independent multi-series forecast since each revenue type had its own unique characteristics and were affected by similar forces.

5.2.1.3. Exogenous Variables

These are predictors common to all series and known at the point of forecasting. The shared variables were derived from the date component of the time series, whose frequency was monthly. Date parts and their respective cyclical features were engineered to help capture seasonality within each series.

5.2.1.4. Lags

These are past values of the target series which are useful in predicting future values. Since the collections are captured with monthly frequency, a lag represents the collection recorded at a point, one or more months in the past. They are also indicative of seasonality in the data.

5.2.1.5. Windows

These are predictors calculated from existing revenue collections by applying a statistic (e.g. mean, median or standard deviation) over past data. Specifically, a rolling window was

generated for each model using a window size that could accommodate the maximum lag while retaining the highest predictive power.

5.2.1.6. Transformers

These are preprocessing steps used to normalize either the target or predictor variables so that they can conform to a standard scale and reduce the effect of outliers on predicted values. The target values were scaled using a custom logarithmic function to prevent negative predictions, while the exogenous variables were scaled using an instance of *RobustScaler* that relies on the median and the interquartile range.

5.3. Model Implementation

A recursive strategy for multiple independent time series was used to implement the different machine learning models. The recursive approach means that a single model is trained to predict one step ahead, then this model is used to predict as many steps into the forecasting horizon as required. The multiple independent time series are the unique revenue streams that make up the County's revenue collected during the period under study. These are treated as individual series with their own patterns. However, they still share common properties that transcend any one stream, and can be learned by the model. The model therefore has the advantage of being a composite entity that can generate multiple predictions without the series acting as predictors between themselves, as in the case of multivariate forecasting.

5.3.1. Exploratory Data Analysis

The purpose of this step was to get an understanding of the data set obtained from Nairobi City County during data collection. It involved analyzing the rows and columns and discovering their data types, frequency distribution and descriptive statistics. In summary, there were eight hundred and forty (840) rows and six (6) columns, from which one column contained a monthly date-stamp for a time-indexed analysis. The target column for prediction was the actual monthly collections recorded per revenue stream. Summary statistics such as the measures of central tendency, generated from the dataset, are shown in Table 5.1. The binned quarterly collections over the period under study are displayed in a histogram in Figure 5.2.

Table 5.1: Summary Statistics for the Dependent Variable

	MonthlyCollection							
	count	mean	std	min	25%	50%	75%	max
RevenueType								
Billboards & Advertisement	60.00	46620219.13	36217827.38	0.00	22326975.00	39745516.50	69802881.50	144051174.00
Decentralisation	57.00	1148844.16	1068568.67	0.00	275261.00	912741.00	1774927.00	4794470.00
Education Fees	52.00	22607.94	22375.54	0.00	5000.00	14500.00	42250.00	80000.00
House and Stall Rent	49.00	15850433.29	23933650.58	0.00	850.00	4738415.00	17684078.00	87186452.00
Land Rates	60.00	35766782.37	100247228.97	0.00	1327932.50	2050522.50	7155239.50	568264599.30
Market Fees	60.00	17357036.43	14205568.25	0.00	4015902.00	15351099.00	26559368.25	58556112.00
Mortuary Fees	59.00	2641345.29	2266880.65	0.00	435550.00	2779401.00	3676977.50	12376358.00
Other Income	60.00	164677961.20	567873010.92	0.00	21282548.75	34342414.50	71511995.50	3444340243.00
Parking Fees	60.00	47953841.27	73976295.39	0.00	2335587.25	5872910.50	63051062.50	280125062.10
Plan Inspection and Approvals	60.00	46215427.74	37398150.97	0.00	10098815.00	41774871.00	70662028.25	130215726.00
Rent	45.00	136122.64	403564.14	0.00	600.00	42400.00	166100.00	2700000.00
SDR	60.00	4991024.97	5628834.64	0.00	450635.00	4261745.00	6200183.75	27346539.00
Single Business Permits	60.00	260321645.92	261576032.37	0.00	86919003.25	183471578.50	331547380.00	1052167641.00
TPS	60.00	1846847.10	2325327.97	0.00	433388.25	940209.50	2570224.25	11523699.00

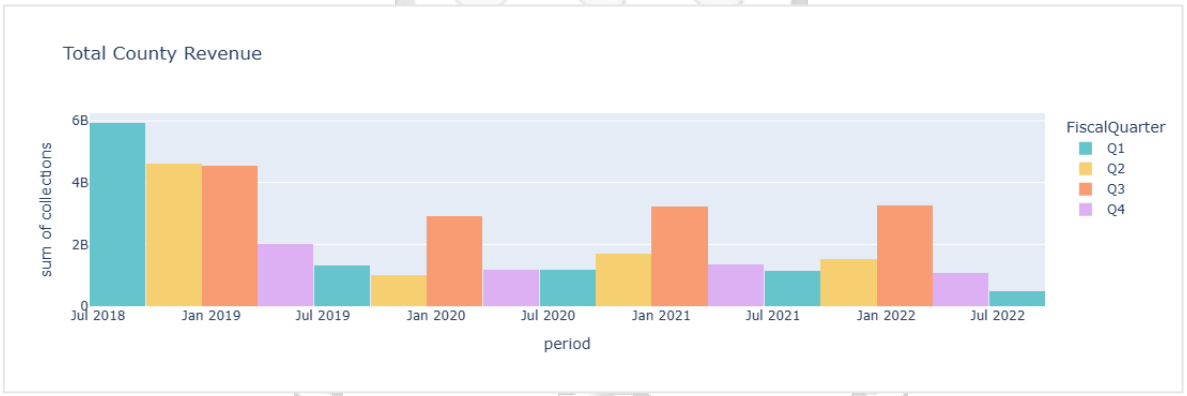


Figure 5.2: Total NCCG Revenue by Fiscal Quarter

Furthermore, the monthly collections were broken down into fourteen categories representing the main sources of revenue. This is not to say that there are no other sources of revenue at Nairobi City County, but these account for approximately 80% of collections and can give a fairly good estimate of future income. These categories are shown in Figure 5.3.

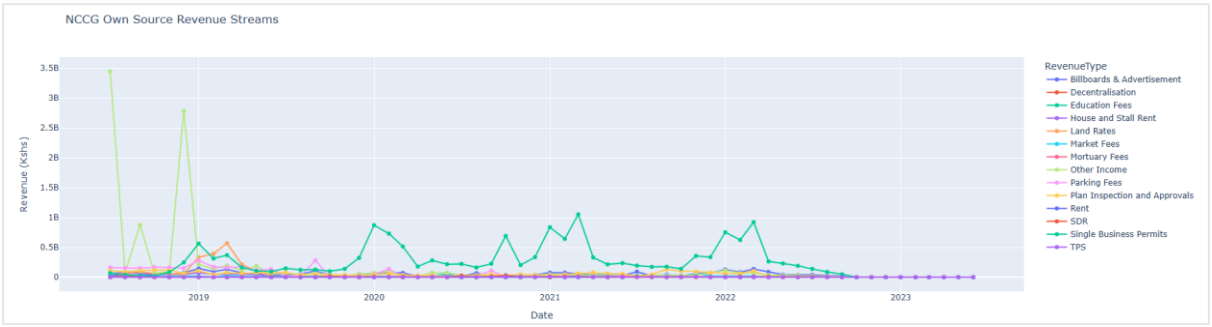


Figure 5.3: Individual OSR Stream Collections Over Time

5.3.2. Data Preprocessing

The data was checked for null values and outliers which could impact the models built. The records spanned from July 2018 to June 2023, which represented five fiscal years. However, there was missing data after September 2022, perhaps due to a recording error. Null values at the tail end of the data set were dropped and those within the time series were imputed via linear interpolation. Two methods were then used to detect outliers. Initially, this was done by use of a box plot, which is able to visualize data points that fall outside the inter-quartile range centered on the median of the distribution. The graph is shown in Figure 5.4.

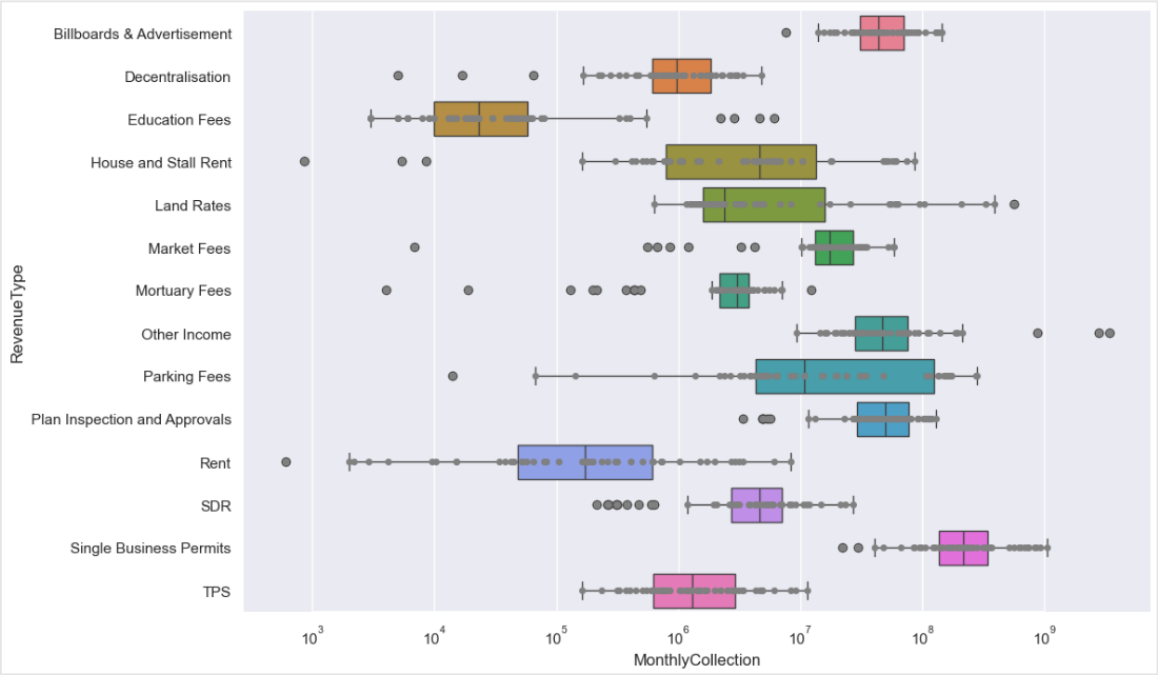


Figure 5.4: Outliers Depicted by a Box and Whisker Plot

Secondly, a time series comprises of properties like seasonality, trend and residual noise. These properties are embedded in the data and were extracted via a Seasonal-Trend decomposition using LOESS (Locally Estimated Scatterplot Smoothing). This technique was able to isolate the respective properties and plot them individually for further analysis. The residual plot was able to identify which data points were likely to be outliers. A sample plot of one of the revenue streams is present in Figure 5.5.

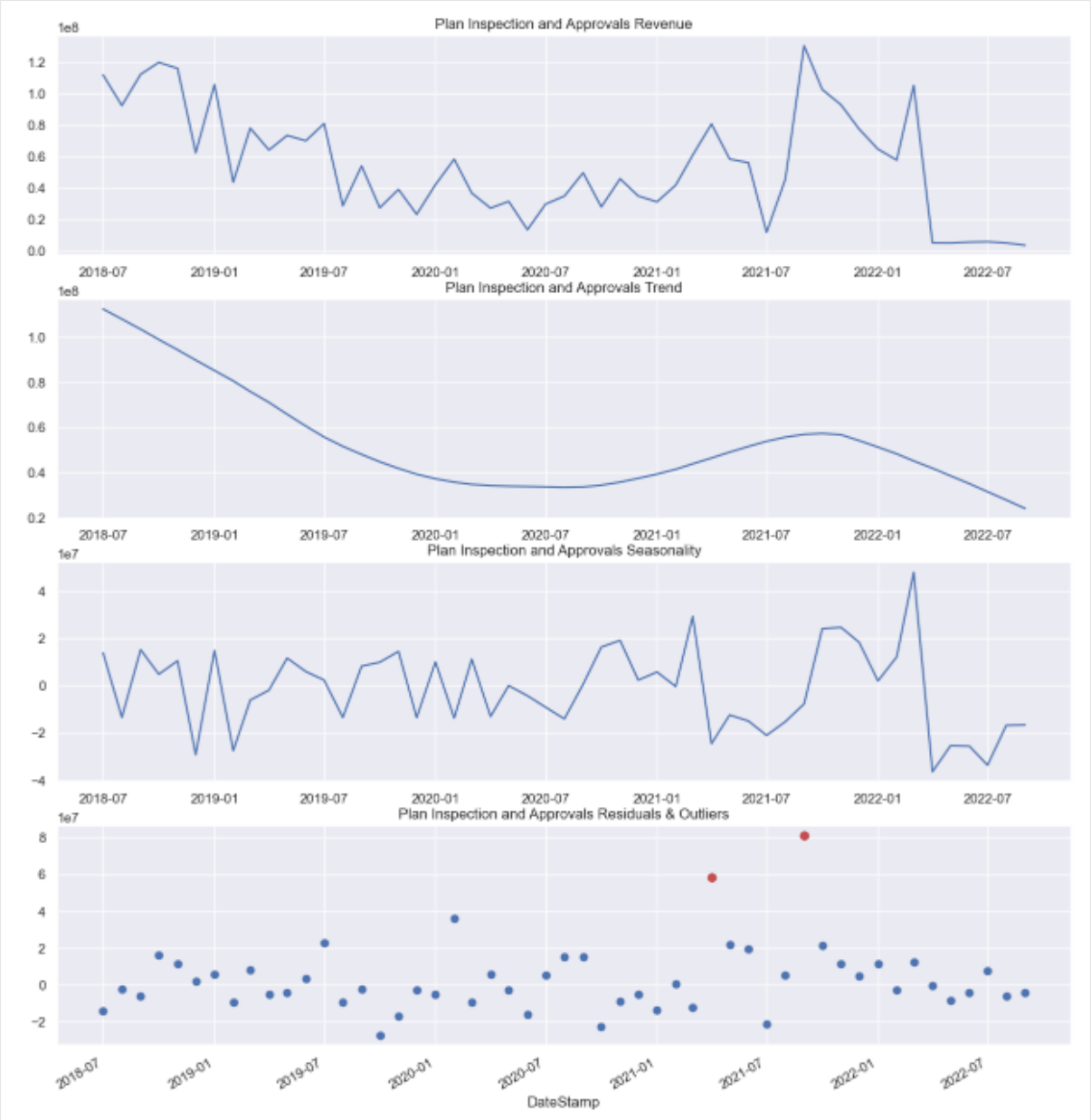


Figure 5.5: STL Decomposition of a Single Revenue Stream (Plan Inspection and Approvals) to Extract Its Seasonality, Trend and Residual Noise

The outliers were highlighted amongst residual values as those data points which lay beyond the upper and lower limits of the inter-quartile range. Each revenue type had its own dynamics and further proved that patterns in the data were unique to each stream. The next step was to identify the periodicity and stationarity of the data. This was done by generating Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots for each revenue type against lagged versions of themselves. Samples are represented in Figure 5.6 and Figure 5.7.

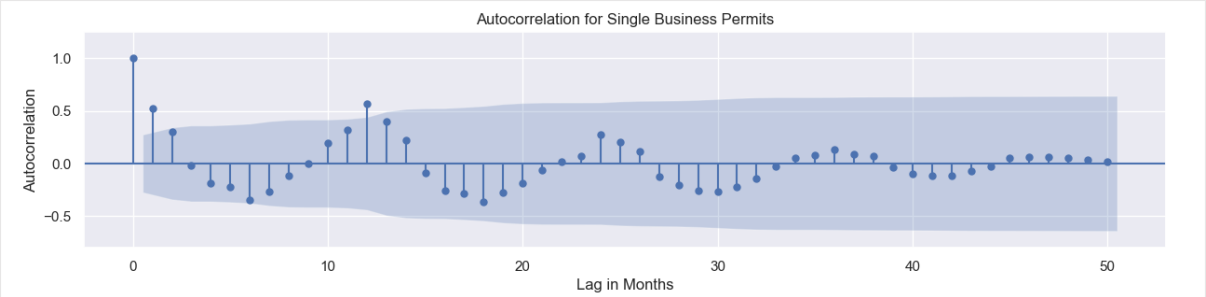


Figure 5.6: ACF Plot for Single Business Permits

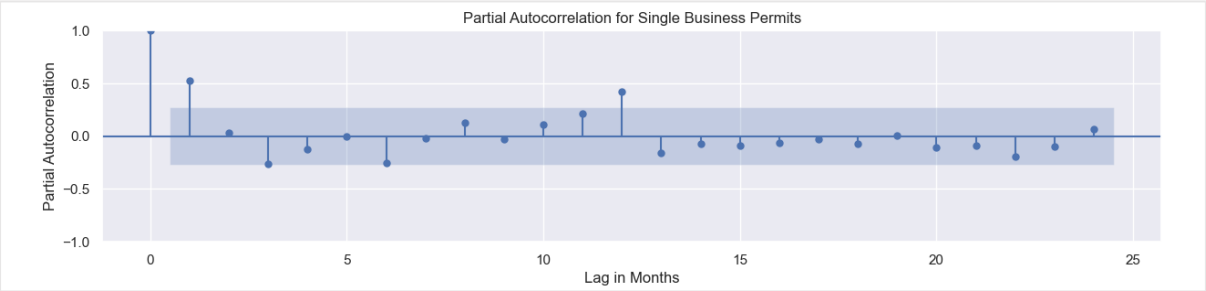


Figure 5.7: PACF Plot for Single Business Permits

These plots showed a general seasonality of twelve months in the data, with shorter lags of five months or less, being more statistically significant.

5.3.3. Feature Engineering

In this step, additional features were added to the data set to enrich its predictors. These comprised of date-time features and their cyclical equivalents. They were generated through a pipeline and transformers, and the numerical outputs were listed as exogenous variables for use in the regression models.

5.3.4. Forecasting

The data was sorted by its date-time index, then split into train, validation and test data sets in a ratio of 36 months (70.6%), 9 months (17.6%) and 6 months (11.8%) respectively. This split was done prior to training to avoid the risk of data leakage, whereby values that lie ahead of the forecast point are seen by the model, which could give great performance scores yet have poor generalization on unseen data (Lones, 2024). The exogenous variables from feature engineering were based on the index, which is predictable and common to all revenue streams. Lags and windows on the other hand were used retrospectively.

Feature selection was also done via Recursive Feature Elimination with Cross-Validation (RFECV) and Sequential Feature Selector (SFS). These techniques were able to perform dimensionality reduction on the lags, windows and exogenous features, thereby simplifying the model inputs. The advantage of this is that models are more robust and avoid overfitting the training data (Galli, 2022). Seven different models were implemented in total, which constituted one statistical model, and six machine learning models.

5.3.4.1. SARIMAX

SARIMAX (Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors) is a statistical model which was used to generate baseline predictions that could be compared with other ML-based models. It is important to note that the model could not be trained on all independent revenue streams simultaneously, and therefore the problem space was simplified by performing predictions on the total NCC revenue as a single series. It could however accept exogenous variables, which were used by the other models. The best parameters found were in the order of (12, 1, 1); meaning an autoregressive component of 12 which follows a yearly seasonality for past values, a differencing component of 1 for stationarity, and a moving average component of 1 as a linear relationship with residual terms. The seasonal order was (0,0,0,0) meaning no order was used for seasonal components.

5.3.4.2. Ridge Regression

Ridge regression is a linear model that employs L2 regularization and addresses multicollinearity in data. It reduces the model coefficients through multiplication by a penalty term, and this is controlled through the alpha parameter. In this experiment, the alpha value was reduced from 1 to 0.304, meaning the penalty was reduced and therefore the magnitude of coefficients increased. This parameter was captured in the model's forecaster object.

5.3.4.3. LightGBM

Light GBM is a tree-based regressor that minimizes a loss function through a series of decision trees that each reduce an error from the previous tree at the leaf level. Several parameters were specified for this algorithm and later tuned to return the best predictions. Boosting was done via the Gradient Boosting Decision Tree (GBDT) method, with a learning rate of 0.1.

5.3.4.4. Random Forest

Random Forest is another tree-based regressor made up of an ensemble of multiple decision trees that randomly split the data into subsets until they arrive at the best estimate that

minimizes the objective function, and then apply bootstrap aggregation (bagging) to average out the results. In this particular application, the branching criterion used was the absolute error.

5.3.4.5. K-Nearest Neighbor

K-NN Regression is a distance-based regressor that uses the target's proximity to the training data in order to determine its prediction by averaging their values. While modeling this algorithm, the optimum number of neighbors was found to be five, and the distance metric used was the Euclidean distance, or Minkowski value of p=2.

5.3.4.6. Support Vector Regression

Support Vector Machine Regression is a non-linear estimator that maps input features to output values while minimizing the prediction error. It does so by using a parameter epsilon to determine the hyperplane margin's width, and kernel functions to transform data into a higher-dimensional space. In this study, a polynomial kernel of degree 2 was found to best hyperparameter to use on the model.

5.3.4.7. Recurrent Neural Network

Recurrent Neural Networks are ANNs better suited for time series data by keeping a history of previous elements in a sequence as a hidden state, to influence the current output. In this experiment, the Keras Functional API was used under the hood to allow for multiple inputs and outputs, as necessitated by the multiple time series in the data set. Two dense and two recurrent layers with Long Short-Term Memory (LSTM) were specified, and a Rectified Linear Unit (ReLU) as the activation function. Fifty epochs were run with early stopping if there was no further decrease on the validation loss after some steps. A comparison of both trends is displayed in Figure 5.8.

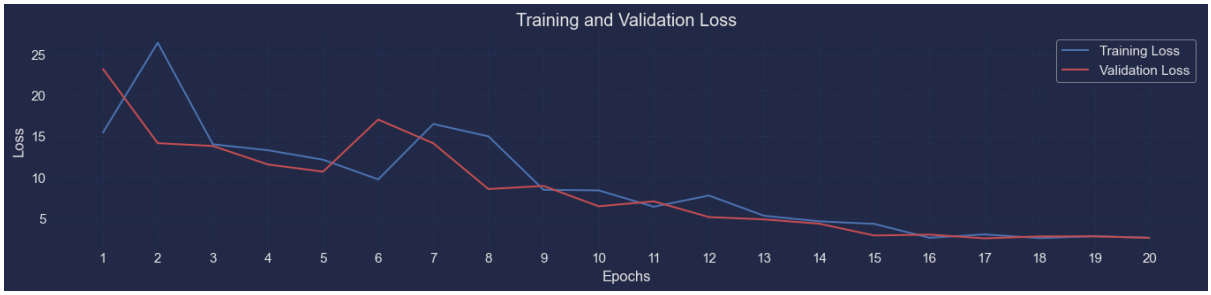


Figure 5.8: Monitoring the Training and Validation Loss of the RNN Forecaster to Prevent Overfitting

5.4. Model Validation

Backtesting with cross-validation over the validation data set was performed after initial model training. This was done to measure the performance of the trained models on historical data. The models were refitted on the data set after each fold, whereas the training set size expanded by a fixed number of steps at each step. Figure 5.9 shows an output of this process over the training and validation sets.

```
Information of folds
--------------------
Number of observations used for initial training: 24
Number of observations used for backtesting: 21
    Number of folds: 7
    Number skipped folds: 0
    Number of steps per fold: 3
    Number of steps to exclude between last observed data (last window) and predictions (gap): 0

Fold: 0
    Training:   2018-07-01 00:00:00 -- 2020-06-01 00:00:00  (n=24)
    Validation: 2020-07-01 00:00:00 -- 2020-09-01 00:00:00  (n=3)
Fold: 1
    Training:   2018-07-01 00:00:00 -- 2020-09-01 00:00:00  (n=27)
    Validation: 2020-10-01 00:00:00 -- 2020-12-01 00:00:00  (n=3)
Fold: 2
    Training:   2018-07-01 00:00:00 -- 2020-12-01 00:00:00  (n=30)
    Validation: 2021-01-01 00:00:00 -- 2021-03-01 00:00:00  (n=3)
Fold: 3
    Training:   2018-07-01 00:00:00 -- 2021-03-01 00:00:00  (n=33)
    Validation: 2021-04-01 00:00:00 -- 2021-06-01 00:00:00  (n=3)
Fold: 4
    Training:   2018-07-01 00:00:00 -- 2021-06-01 00:00:00  (n=36)
    Validation: 2021-07-01 00:00:00 -- 2021-09-01 00:00:00  (n=3)
Fold: 5
    Training:   2018-07-01 00:00:00 -- 2021-09-01 00:00:00  (n=39)
    Validation: 2021-10-01 00:00:00 -- 2021-12-01 00:00:00  (n=3)
Fold: 6
    Training:   2018-07-01 00:00:00 -- 2021-12-01 00:00:00  (n=42)
    Validation: 2022-01-01 00:00:00 -- 2022-03-01 00:00:00  (n=3)

  0%|          | 0/7 [00:00<?, ?it/s]
--------------------
```

Figure 5.9: Cross-validation with an Expanding Window Size and Refitting at Every Step

Individual plots were made after this process to show how each model fared when predicting on the validation data set. This helped to visualize how closely the regressors estimated the actual revenue collected over that period. Error metrics were also collected to quantify the individual model backtesting performance. Figure 5.10 shows individual model performance.

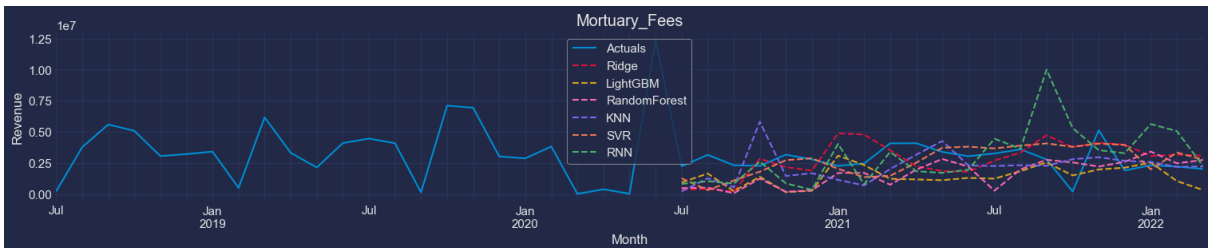


Figure 5.10: Model Backtesting Performance on the Mortuary Fees Revenue Stream

Hyperparameter tuning was carried out using the Bayesian Search method for the machine learning models and Grid Search for the statistical model. The objective function was to

minimize the Mean Absolute Scaled Error metric, while identifying and refitting the forecasters with the best lags and model hyperparameters found, for every level or series of revenue stream in the data set. The only exception was the RNN model, whereby a suitable Keras compatible tuner that could accept the forecaster object's structure and Keras functional API type of combination was not found.

5.5. Model Testing

Probabilistic forecasting was done whereby testing results were obtained for each model using prediction intervals of 95%. This took into account the upper and lower 2.5% percentiles to give a range within which the predicted values were likely to fall. This was useful when superimposed against the actual values in the test data to see overlaps, and the certainty with which a regression line was chosen. These forecasts were then plotted for all models and all revenue streams. Due to the singular choice of target for SARIMAX, its plot represented the entire revenue as a series and not any particular stream. Figures 5.11, 5.12 & 5.13 depict predictions and likelihoods from different models plotted against the test data set.

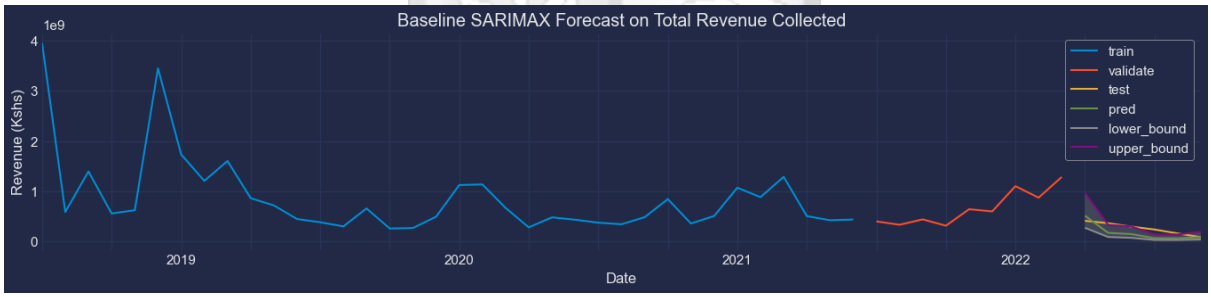


Figure 5.11: A Line Plot of SARIMAX Predicted Values Compared to the Full Data Set

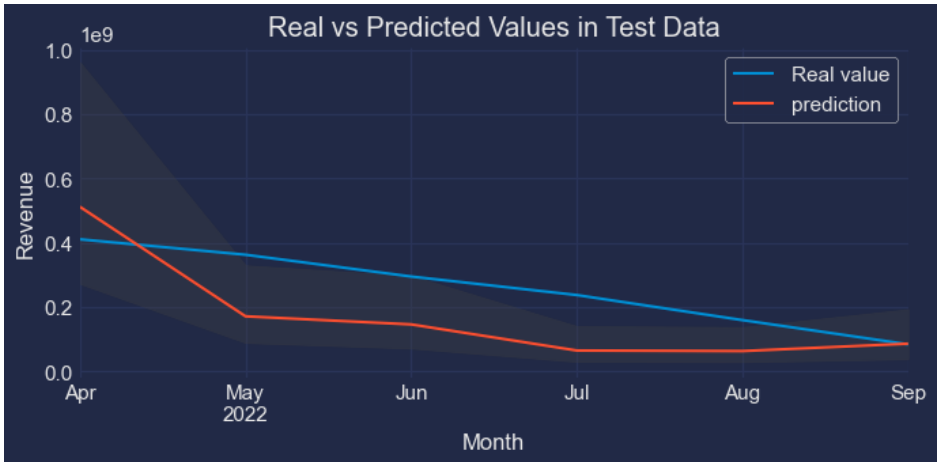


Figure 5.12: A Focused View of the SARIMAX Prediction Interval on the Test Data

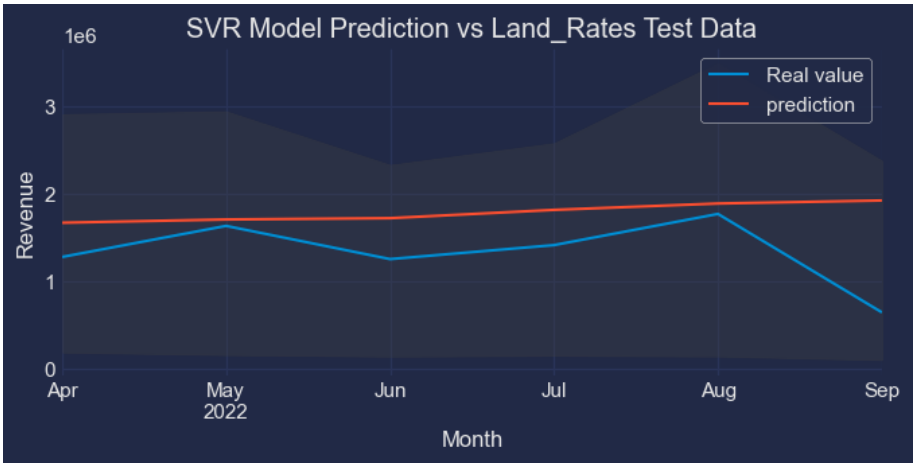


Figure 5.13: Support Vector Regression (SVR) Prediction Interval on Land Rates Test Data

Five error metrics were collected per model at this stage, that is the Mean Absolute Scaled Error (MASE), Weighted Absolute Percentage Error (WAPE), Forecast Bias, R^2 Score and Accuracy. MASE and WAPE were preferred since these are scale independent, less sensitive to outliers and easier to interpret given the scale of predictions (typically in the millions) and their residuals, which vary from one revenue stream to another. The Forecast Bias helps to gauge whether the model is over or under forecasting its predicted values relative to the actual figures. The R^2 score meanwhile shows how much of the variation in the target values is accounted for by the predictors. The model accuracy is defined as 1-WAPE and expressed as a percentage. Metrics collected from the machine learning models were aggregated by averaging across all revenue streams to get a general idea of their performance. Scores from the statistical model were also gathered for comparison and benchmarking of results. The average values of metrics scored across all models are captured in Table 5.2.

Table 5.2: Aggregated Error Metrics from Model Testing

	SARIMAX_Total	Ridge_Average	LGBM_Average	RF_Average	KNN_Average	SVR_Average	RNN_Average
MASE	0.2329	0.8228	1.2378	0.8365	1.0317	1.5807	0.8453
WAPE	0.4602	1.0876	2.0709	1.2270	1.5903	2.6195	1.2613
Forecast_Bias	0.3268	-0.4215	-1.6962	-0.8436	-1.3032	-2.3883	-0.7905
R2_Score	-0.4210	-155.6198	-403.8886	-204.6770	-376.0760	-879.0531	-0.6838
Accuracy	53.9826	-8.7642	-107.0946	-22.7046	-59.0253	-161.9476	-26.1264

5.6. Web Application

After model development was complete, a web application was created to allow users to interact with it and run forecasts based on available data. The three functional requirements specified in section 4.2.1 were implemented in the app as core features and laid out in a clean and simple user interface.

5.6.1. Back-end Environment

The environment that was used to develop and test both the models and web application comprised of the technologies shown in Table 5.3.

Table 5.3: Hardware and Software Environment Specifications

Hardware	**Software**
Intel Core i7-8650U CPU	Windows 11 Pro (x64)
16 GB DDR3 RAM	PostgreSQL 16rc1
Intel UHD Graphics 620	Python 3.9.7
1TB SSD NVMe Disk	Flask 2.2.5
	Skforecast 0.14.0
	Plotly 5.24.1
	Vizro 0.1.38

5.6.2. Front-end Interface

The user interacts with the application via their web browser. A set of pages are available to help them achieve respective tasks.

5.6.2.1. Login Page

Users are required to log into the application in order to access its functionality. This is done via the login link on the menu. In case the user has no account previously set up, they can create a new one by using the registration page. These login details are stored in a database table and sensitive information like passwords are hashed to maintain privacy and security. An example of this login page is shown in Figure 5.14.

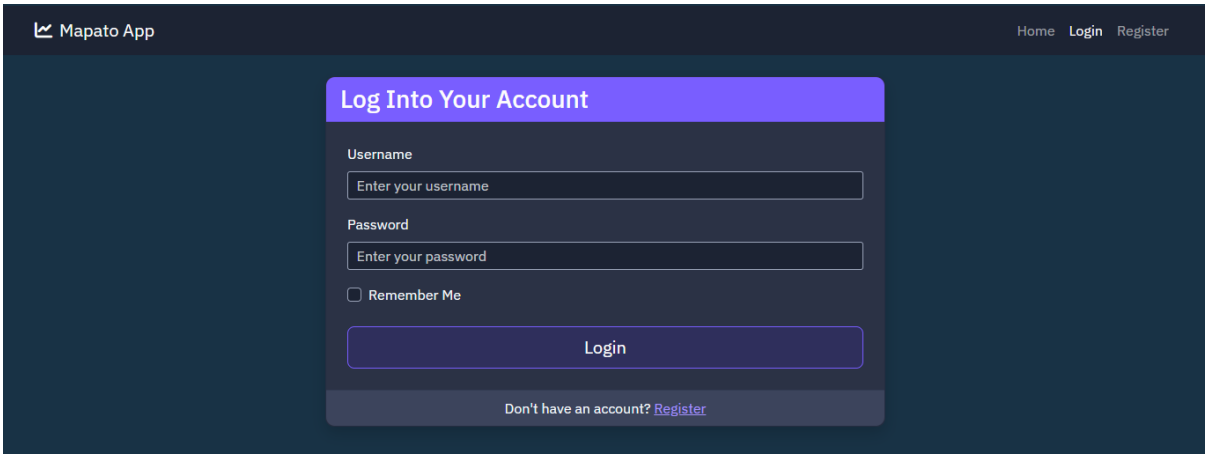


Figure 5.14: Login Page

5.6.2.2. Profile Page

Once the user logs in, they are presented with their profile details which includes their assigned role, username, email and relevant date stamps. This is displayed in Figure 5.15.

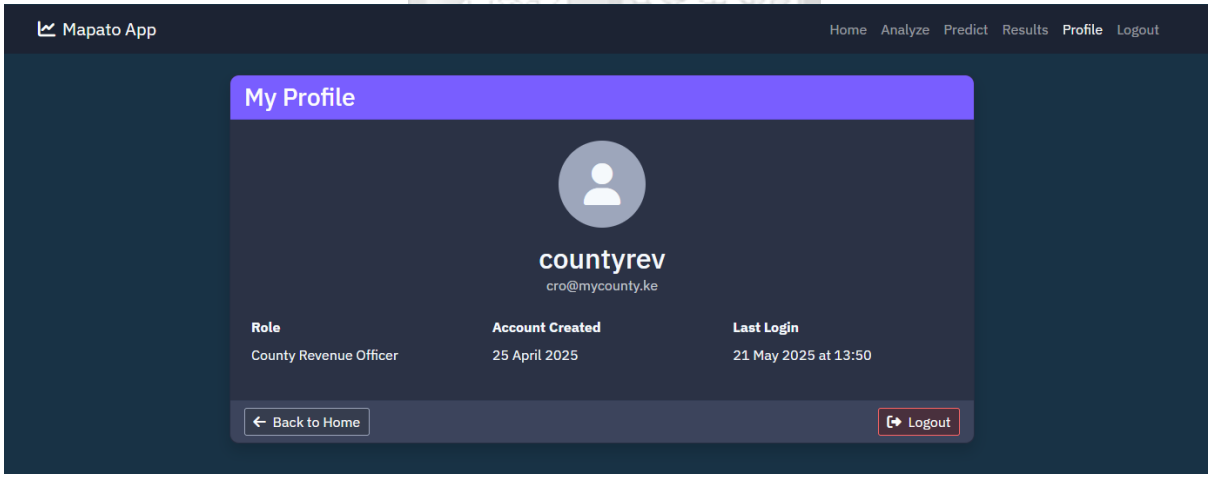


Figure 5.15: User Profile Page

5.6.2.3. Home Page

Figure 5.16 illustrates the main home page. This is the starting point when testing out the application. It gives a brief summary of what the app aims to do, so that the user is guided on what the menu items mean and what they can expect to do after visiting those pages.

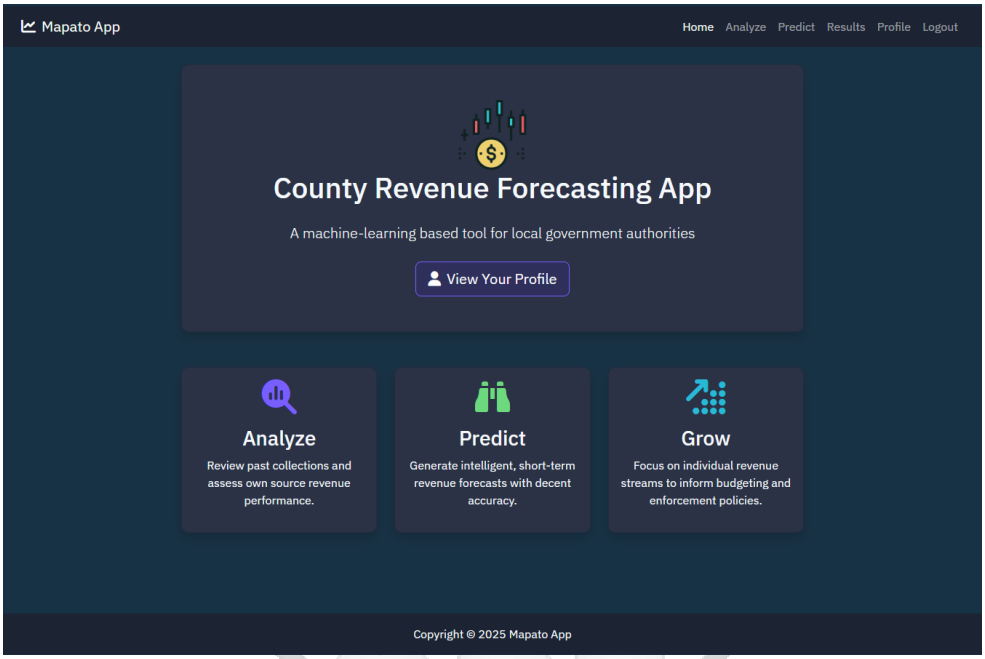


Figure 5.16: Home Page

5.6.2.4. Analyze Page

The analyze page shown in Figure 5.17 contains an interactive dashboard through which users can sift through the entire dataset and visualize analyses by filtering on columns and ranges.

Figure 5.17: Revenue Analysis Dashboard

5.6.2.5. Predict Page

The predict page in Figure 5.18 enables users to select parameters that will be needed to serve predictions. These include the revenue stream, forecast point and horizon, and its probability.

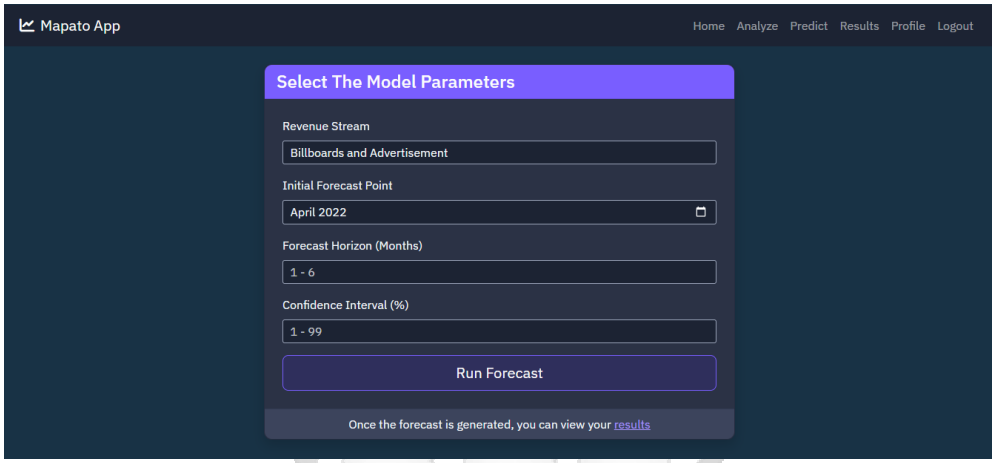


Figure 5.18: Prediction Input Page

5.6.2.6. Results Page

After the forecast is run, the generated results are passed to another page for visualization and inspection. These are captured on a static table and interactive graph as shown in Figure 5.19.

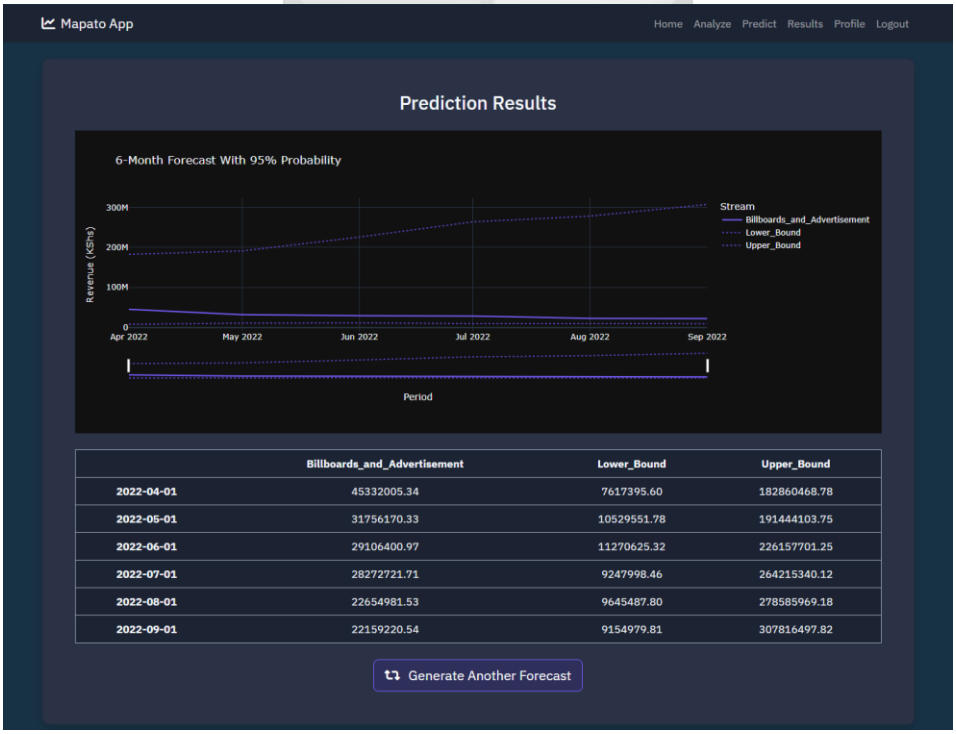


Figure 5.19: Forecast Results Page

6. Discussion

6.1. Introduction

The aim of this study was to develop a machine-learning model for revenue forecasting at local government authorities in Kenya, with a keen interest in Nairobi City County as a pilot site. This was done to offer a scientific approach towards the generation of forecasts using historical data, so that their accuracy could be increased, estimates could be factual and realistic, and the county's budgeting process could be enhanced. The challenge was tackled through extensive research into methods used in this domain and armed with this knowledge, data was collected from the county for use in building the proposed model. The study was able to explore various ML based techniques and examine their performance levels, and therefore their suitability for this purpose. This section delves into the results obtained from this process and identifies candidate models for adoption by county governments.

6.2. Expected and Unexpected Results

The research was able to evaluate different models and contrast this with findings published in literature. According to CEPAL (2015) and Chung, et al., (2022), ARIMA method gives the highest forecasting accuracy amongst non-machine-learning techniques. This was found to be factual in this study, despite only one variant, SARIMAX, being used for statistical forecasting. A like for like comparison could not be done between it and other machine learning models, which had the superior ability to accept multiple inputs and churn out multiple predictions to cover individual revenue streams. However, the ML model results in this research showed their potential for application in this area.

Hajek and Olej (2010) suggest that ANN and SVM ensembles with bagging and boosting outperform either a single Neural Network or SVM in its prediction accuracy, and that this holds true for regression problems. A feed-forward neural network was used as a base learner and predictions pooled with those of SVM to get a consensus on the collective prediction. In this particular research, RNN and SVR were used to generate separate predictions without further use in an ensemble. There was however a model whose nature it was to treat a series of decision trees as an ensemble, and that is the Random Forest model. This yielded one of the better MASE scores of the models evaluated at 0.8365, however the drawback to this ensemble

technique was that it was quite slow. They concluded that SVM and FFNN model non-linear relationships well and are appropriate for use with municipal revenue in the Czech Republic.

Similarly, Yang, et al., (2018) posit that using principal component analysis on a genetic algorithm SVM performs better with a MAPE of 6.01% over a neural network error of 19.39% for public fiscal revenue prediction in Harbin City, China. In this study with NCC data, results were contradictory with RNN outperforming SVR when predicting on the test data set with WAPE scores of 1.88 vs 2.09 respectively. Fikriya & Hikmawati (2020) corroborate the use of SVM in their study of Tax revenue in South Lampung, Indonesia. They relied on a polynomial kernel and were able to predict hotel revenue tax at about 85% of the actual collection for the first half of the year 2019. In comparison, the highest forecast accuracy for NCC data was obtained with land rates at 79.8% whereas SVR performed better than the rest only with mortuary fees at 54.1% accuracy.

6.3. Interpretation and Implication of the Results

The first research objective was to investigate the challenges associated with revenue forecasting. These factors are highlighted in section 2.1 as uncertainty in the socio-political environment within which revenue is collected, purposeful underestimation in order to meet set targets, and a lack of expertise in quantitative methods by government officials. The consequences of these challenges are the existence of unrealistic budgets, which often result in perennial shortfalls that impact expenditure on development programs and service delivery to citizens. It also leads to over-reliance on external funding like donor grants and equitable share from national government, as opposed to local authorities maximizing efforts in gathering own source revenue. One of the respondents interviewed at Nairobi County cited the unavailability of data especially for unstructured revenues as a key hindrance to the generation of OSR forecasts by line departments.

The second research objective was to evaluate methods, techniques and approaches used in forecasting revenue collection. This was further elaborated in section 2.2, with the key differences being whether the methods were qualitative or quantitative in nature. Machine learning techniques were discussed in section 2.3 and an assessment of pros and cons to each approach carried out in section 2.4. No assumptions were made on the suitability of any one method, rather an appreciation for their relevance was made with an openness to experiment.

The third research objective was to formulate a model to forecast revenue collection using machine learning algorithms. Section 2.5 gives an overview of how this process would be achieved. The actual models assessed were seven in total, and these are captured in section 5.3.4 with descriptions of their general architecture and specific choice of parameters as pertains to this revenue forecasting problem.

The final objective was to validate the proposed model. To this end, the developed models were used to make predictions against a test data set and error metrics gathered to evaluate their performance. A trade-off was to be made between model accuracy, speed and interpretability. To ensure a fair appraisal, an average metric was calculated to represent performance across all revenue types. When the average MASE value is greater than one (1), it indicates that the algorithm is performing poorly compared to a naïve forecast (which is simply equated to the previous time step in the time series). SARIMAX, Ridge and Random Forest models were found to outperform such a forecast. However, other models scored a MASE above one, and therefore trail behind when compared to predicting a lag of one timestep in the past. This result is shown in Figure 6.1.

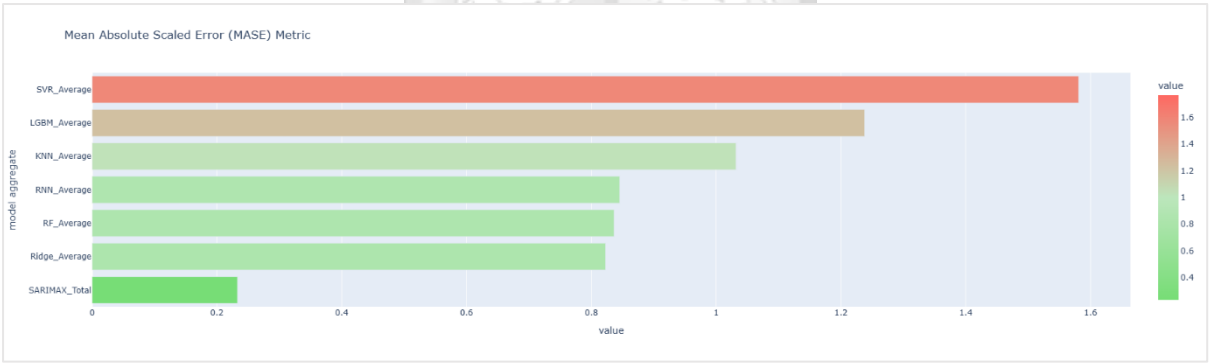


Figure 6.1: MASE Results from Model Prediction on Test Data

A lower WAPE value indicates a more accurate model. During evaluation, SARIMAX returned the best WAPE accuracy score of 0.4602. Higher WAPE scores above one (1) implied a greater deviation of forecasted values from actual values i.e. a higher error rate. This could be attributed to Parking Fees and Plan Inspection and Approvals scores which greatly skewed all model averages. The sub-par scores in these two streams could indicate that there might be other external drivers which should be factored in when creating predictor variables, so that the models may learn complex patterns about such phenomena. The scores are shown in Figure 6.2.

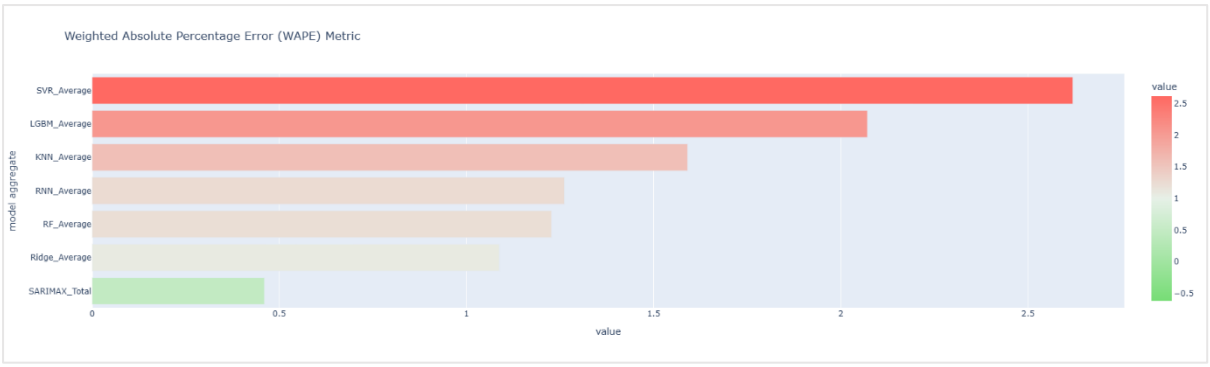


Figure 6.2: WAPE Results from Model Prediction on Test Data

When the average forecast bias is negative, the respective model consistently over-forecasts for that revenue type. The inverse is that when the average forecast bias is positive, that model consistently under-forecasts for that revenue type. Most of the Machine Learning models seemed, at least on average, to over-forecast the actual values in the test data set. SARIMAX is the only model that under-forecasted the singular target (total revenue), though not by much at a score of 0.3268. The best performer from the group of machine learning models was the Ridge Regressor, which managed to over-forecast with a score of -0.4215. The different biases are displayed in Figure 6.3.

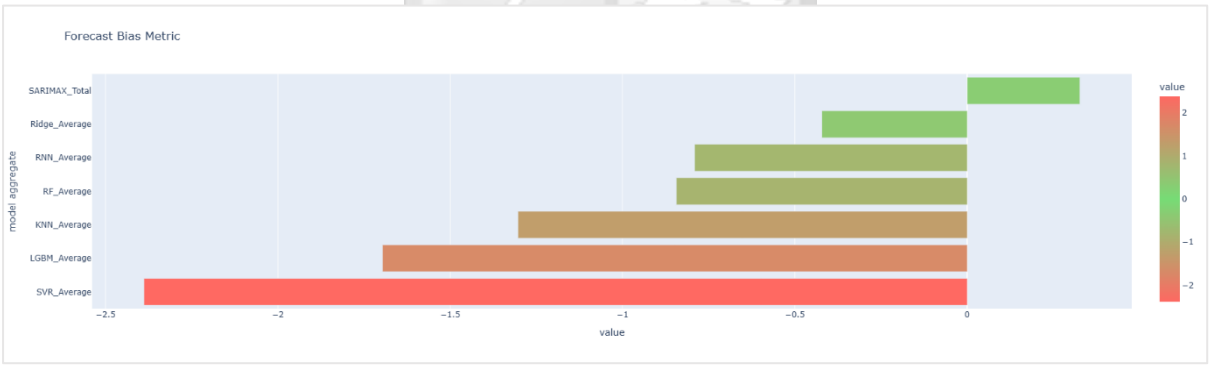


Figure 6.3: Forecast Bias Results from Model Prediction on Test Data

The R-squared value typically ranges from 0 to 1. The negative values returned across all models imply that the fitted models are either non-linear, don't align with trend in the data, or don't generalize well on data other than that used to train them. SARIMAX returned the best score of -0.421 with RNN coming close at -0.6838. Given the two streams mentioned earlier, the SVR model was particularly impacted when predicting Plan Inspection and Approvals. It may require further hyperparameter tuning to get the best regression line that fits a majority of data points. The R^2 metric for each model is shown in Figure 6.4.

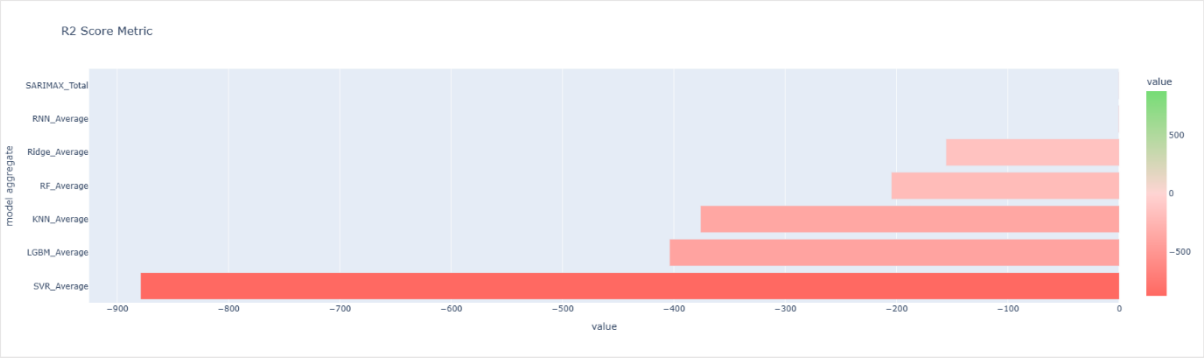


Figure 6.4: R-Squared Results from Model Prediction on Test Data

As far as model prediction accuracy goes, there were mixed results across the board. A benchmark of 53.9% was set by the statistical model on the test data set, though this was achieved on account of the sum of all revenue streams as a single series. The drill-down in figure 6.5 shows individual revenue stream performance, and the models that achieved the best scores in each series. It is important to note that machine learning models beat this benchmark in six out of fourteen streams, with Random Forest featuring in five series and getting the highest overall score of 79.8% with the Land Rates revenue stream; while Ridge was a close second with a leading score in four series and the second highest individual score of 74.3% accuracy, as seen with the Other Income revenue stream. Just as it was picked up by other metrics, two series grossly underperformed, that is parking fees and plan inspection and approvals, both with negative scores that are emblematic of overfitting in previous training and validation stages, or the presence of outliers that still need to be addressed. These individual revenue stream accuracies are present in Figure 6.5.

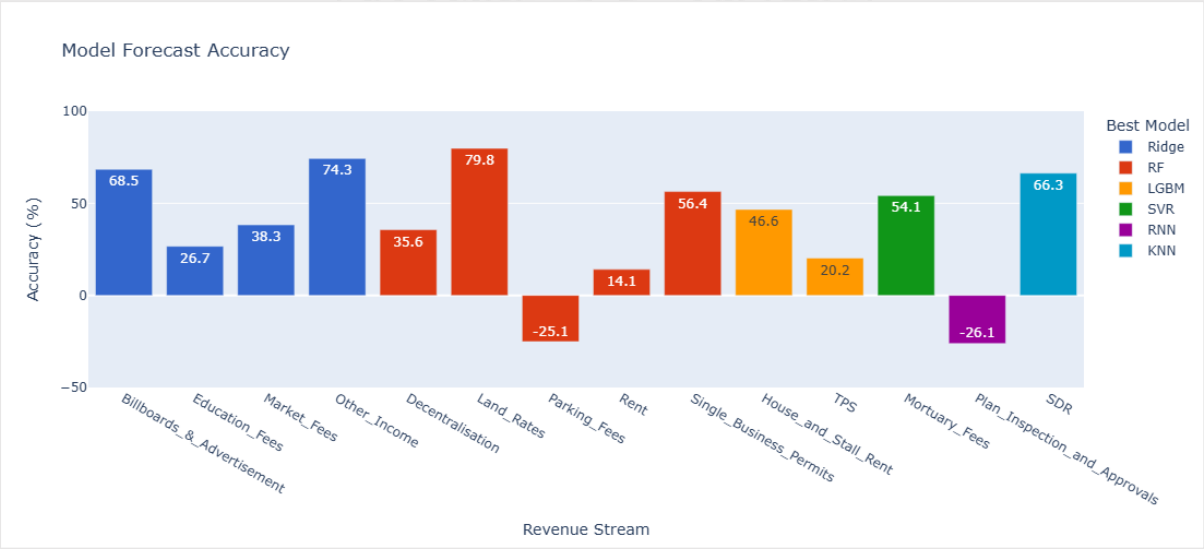


Figure 6.5: Forecast Accuracy of the Best Performing Models in Each Revenue Stream

The final metric observed from the models was their execution time at different stages of implementation. The average was 2.4s for validation, 38.5s for hyperparameter tuning and 32.8s for testing. The fastest performers were Ridge during validation at 0.3s, KNN during tuning at 3.3s and SARIMAX during testing at 0.1s. The least computationally intensive model was SVR with a total runtime of 4.7s for all three stages, while the most intensive model was Random Forest which peaked at 276.5s. This variation may be down to the inner workings of the RF algorithm, which is an ensemble of decision trees, and therefore each step adds more complexity to the computation. It should also be noted that these tests were run on a CPU-based platform and there could be some performance gains to be made with the use of GPUs. Figure 6.6 shows the speed of each model relative to one another.

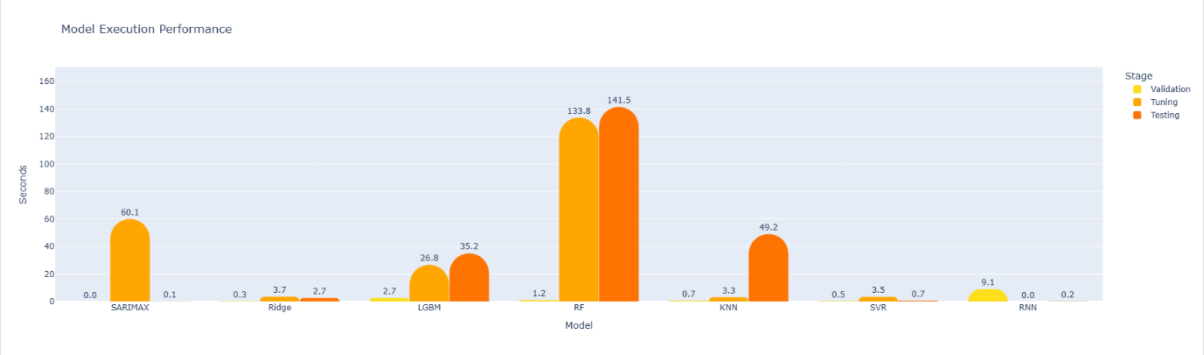


Figure 6.6: Model Runtimes at Each Stage of Implementation

6.4. Summary

The outcomes of this experiment show that machine learning algorithms can produce forecasts that are comparable to benchmark methods, and in some cases, outperform them at individual series prediction. Caution must be taken in interpreting this result, since data quality plays a key role here, as does the sample size. Longer series could help obtain a clearer distribution from the real world, as non-stationarities can affect how models generalize beyond the training set (Hewamalage, Ackermann, & Bergmeir, 2023). The emphasis for local government officials is to consistently capture data on revenue collections, and with minimal transcription errors, which in turn gives better historical context within which to predict the future. Gaps or inaccuracies amongst revenue streams will eventually yield sub-optimal results. There must also be periodic training in a production environment as new data is recorded, to help the models learn from changing dynamics at the county. This can include the introduction of model

parameters that help explain natural phenomena affecting revenue collection, should they be found to have high feature importance, yet are not highly correlated with the target variable.

6.4.1. Contribution to Research

All aspects considered, statistical models still offer good forecasting performance as measured in this experiment. Their limitation is in modeling more nuanced scenarios like the one observed with NCC data, whereby multiple revenue streams have their own individual characteristics. This is where machine learning algorithms offer up their strengths in modeling composite series and learning each independently, which could result in better performance at a local level.

The study's results showed that Ridge Regression was the best-ranked ML model that displayed decent results across all metrics. It was the most consistent and could complement SARIMAX in revenue estimation. Its use could greatly enhance the budgeting process at Kenyan counties, whereby factual targets may be set by revenue officers based on historical collections, and OSR performance shortfalls (with their negative impacts) could be mitigated.

6.4.2. Challenges Faced

Firstly, the quantity of data was a challenge since the data set size was considerably small, especially after dropping some missing values. Machine learning algorithms require many samples to train on and it would be advantageous to have more historical records to use. Secondly, the quality of data could also be better by incorporating different types of predictors to help models learn more patterns at a macro level, and to gain a broader view of drivers for revenue collection. Lastly, hyperparameter tuning for the Recurrent Neural Network model was a challenge due to incompatibility with the library framework used. Doing so may have boosted its prediction accuracy.

7. Conclusion and Recommendations

7.1. Conclusion

The general objective of the study was to develop a machine-learning model for revenue forecasting at local government authorities in Kenya. This was done with a view to increase the accuracy of revenue forecasts and provide a scientific basis for the county budgeting process. A study of reports published by the OCOB showed that on average, County Governments in Kenya achieve less than two thirds of their OSR targets (Office of the Controller of Budget, 2022). This can be attributed to arduous enforcement on the ground, low compliance by citizens, inadequate forecasting capabilities of the revenue management systems in use, poor data quality on revenue streams i.e. accuracy, integrity and completeness, as well as skill gaps within revenue departments. Evidently, this has far-reaching implications beyond revenue underperformance, such as over-expenditure, debt, impeded service delivery and reduced capacity for decision making.

This research explored both statistical and machine-learning models used in the econometrics domain and evaluated their performance on real-world data sets. The results showed that indeed, ML can be used to forecast LGA revenue, with relatively good accuracy, and that this can be done for both individual revenue streams, as well as aggregated for total county OSR. Fundamentally, the issue with the current LGA forecasts is that they are too ambitious. This tendency was reflected by the ML models tested and contradicted by the statistical model. However, the degree to which the ML models over-forecasted was more reserved than the reported figures, with the best performing model, Ridge Regression, exceeding actual values by a cumulative total of 38.9% for the six months predicted in the test data set, by estimating revenue collections of 2.15B vs the actual 1.548B Kenya Shillings. In essence, this would have represented a relative performance of 72% against the ML-forecasted revenue for Nairobi City County. For context, NCC's reported performance for the year 2022, where the test data lies, was 47.1% against their annual OSR target (Office of the Controller of Budget, 2022).

This study therefore showed the potential of using machine learning models as a tool for revenue forecasting in Kenyan County Governments, where with the right predictors and good quality data, forecasts that closely align with past performance may be derived and used in the budgetary process. Such models should not be taken as direct replacements for approaches currently in use, but they may instead be used to guide expectations on what is feasible. They may also be updated as new information is gathered and reused in a production environment.

7.2. Recommendations

It is this study's recommendation that modern forecasting capabilities be integrated with County Revenue Management Systems to augment their existing functionality. A collaborative effort between department officers and decision makers at the county can be made to identify further pain points and gaps in the forecasting process, and map out areas where this solution can fit in and add value. Corrective courses of action can also be defined such that these policies may be invoked in case forecasts lie within a predicted range. Examples of such measures are increasing enforcement officers on the ground, closing non-compliant businesses, imposing fines on offenders, incentivizing rate payers through waivers, simplifying the license/permit application procedure etc. This will enhance data-driven decision making and targeted resource mobilization. Furthermore, a domain-specific foundation model for all counties in Kenya may be developed using a broader data set, as seen with efforts from open source initiatives like Lag-Llama (Rasul, et al., 2024), such that any county may use it to forecast their Own Source Revenue.

7.3. Future Work

Certain limitations were faced in this research and they could be considered for further investigation and improvement in the area of study:

i. Differencing can be applied to ensure that the series are stationary. This requires a larger data set since some information may be lost in the process.
ii. Weighted window functions can be used to make predictor features more sensitive to recent changes.
iii. Correcting the forecast bias of the models would increase prediction accuracy. Likewise, using a tracking signal for models deployed to production can issue warnings if the forecast bias crosses a set threshold, and avoid wild estimates.
iv. Obtaining a larger data set size and of better quality, for example by having different kinds of exogenous variables, so as to train models on more samples. Behavioral predictors can help adapt models to external signals like civic education campaigns, waivers, socio-political events etc.
v. An increase in the data's frequency may offer insights into shorter forecast horizons e.g. weekly or daily revenue collection.

References

Agile and Harmonized Assistance for Devolved Institutions (AHADI). (2020, September). *County Revenues*. Retrieved from County Governance Toolkit: https://countytoolkit.devolution.go.ke/county-revenues

Aleryani, A. Y. (2016, March). Comparative Study between Data Flow Diagram and Use Case Diagram. *International Journal of Scientific and Research Publications, 6*(3), 124-127.

Avison, D., & Fitzgerald, G. (2006). *Information Systems Development: Methodologies, Techniques & Tools* (4th ed.). Berkshire: McGraw-Hill Education.

Batóg, B., & Batóg, J. (2021). Regional Government Revenue Forecasting: Risk Factors of Investment Financing. *Risks*, 9, 210.

Becris. (2023, August 22). *Lineal Color*. Retrieved from Flaticon: https://www.flaticon.com/authors/becris/lineal-color

Bell, S. (2009). Experimental Design. In R. Kitchin, & N. Thrift, *International Encyclopedia of Human Geography* (pp. 672-675). Saskatoon, Saskatchewan, Canada: Elsevier Science. doi:https://doi.org/10.1016/B978-008044910-4.00431-4

Bruton, L. (2022, September 8). *What is wireframing? A complete guide*. Retrieved September 12, 2023, from UX Design Institute: https://www.uxdesigninstitute.com/blog/what-is-wireframing/

Campbell, D., & Campbell, S. (2008). Introduction to regression and data analysis. *StatLab Workshop Series*, pp. 1-14.

Centre for Economic Governance. (2018). *Automation for County Own Source Revenue: Gaps, Challenges and Solutions.* Nairobi: Centre for Economic Governance.

CEPAL. (2015). *Revenue and expenditure forecasting techniques for a PER Spending*. Retrieved from The United Nations Economic Commission for Latin America and the Caribbean: http://www.cepal.org/sites/default/files/project/files/annex_3_revenue_and_expenditure_forecasting_methods_for_the_sector_per.pdf

Chindia, E. W., Wainaina, G., Kibera, F. N., & Pokhariyal, G. P. (2014). Forecasting Techniques, Operating Environment and Accuracy of Performance Forecasting for

Large Manufacturing Firms in Kenya. *International Journal of Managerial Studies and Research (IJMSR)*, PP 83-100.

Chung, I. H., Williams, D. W., & Do, M. R. (2022). For Better or Worse? Revenue Forecasting with Machine Learning Approaches. *Public Performance & Management Review, 45*(5), 1133-1154. doi:10.1080/15309576.2022.2073551

Commission on Revenue Allocation. (2018). *County Implementation of Automated Revenue Management Systems.* Nairobi: AHADI.

Delhi University. (2023, September 21). *ER Diagram in DBMS.* Retrieved from University of Delhi: https://www.du.ac.in/du/uploads/departments/Operational%20Research/24042020_E-R%20Model.pdf

Dennis, A., Wixom, B. H., & Roth, R. M. (2012). *System Analysis and Design* (5th ed.). Hoboken, NJ: John Wiley & Sons, Inc.

Engstrom, D. F., Ho, D. E., Sharkey, C. M., & Cuéllar, M.-F. (2020). *Government by Algorithm: Artificial Intelligence in Federal Administrative Agencies.* New York University, School of Law. Administrative Conference of the United States.

Fattah, J., Ezzine, L., Aman, Z., Moussami, H. E., & Lachhab, A. (2018). Forecasting of Demand Using ARIMA model. *International Journal of Engineering Business Management, 10*, 1-9.

Fikriya, A. A., & Hikmawati, S. (2020, April). Support Vector Machine Predictive Analysis Implementation: Case Study of Tax Revenue in Government of South Lampung. *Proceeding International Conference on Science and Engineering, 3*, 323-327.

Galli, S. (2022, August 22). *Feature Selection in Machine Learning with Python.* Retrieved from Leanpub: http://leanpub.com/feature-selection-in-machine-learning

Galli, S. (2022). *Python Feature Engineering Cookbook* (2nd ed.). Birmingham, UK: Packt Publishing Ltd.

George, J. F., & Valacich, J. S. (2016). *Modern Systems Analysis and Design* (8th ed.). Boston: Pearson.

Government Finance Officers Association. (2023, June 22). *Financial Forecasting in the Budget Preparation Process*. Retrieved from GFOA: https://www.gfoa.org/materials/financial-forecasting-in-the-budget-preparation-process

Hajek, P., & Olej, V. (2010). Municipal Revenue Prediction by Ensembles of Neural Networks and Support Vector Machines. *WSEAS Transactions on Computers, 9*(11), 1255-1264.

Hayes, A. (2022, May 12). *Econometrics: Definition, Models, and Methods*. Retrieved from Investopedia: https://www.investopedia.com/terms/e/econometrics.asp

Hewamalage, H., Ackermann, K., & Bergmeir, C. (2023, March). Forecast evaluation for data scientists: common pitfalls and best practices. *Data Mining and Knowledge Discovery, 37*, 788–832. doi:https://doi.org/10.1007/s10618-022-00894-5

Hewamalage, H., Bergmeir, C., & Bandara, K. (2021, March 22). *Global Models for Time Series Forecasting: A Simulation Study.* doi:https://doi.org/10.48550/arXiv.2012.12485

Imandoust, S. B., & Bolandraftar, M. (2013). Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background. *International Journal of Engineering Research and Applications, 3*(5), 605-610. Retrieved from www.researchgate.net

Institute of Public Finance - Kenya. (2020, April 25). *What are some of the challenges counties face in revenue collection and is automation of revenue an end in itself?* Retrieved from IPF-Kenya: https://ipfkenya.or.ke/2020/04/25/what-are-some-of-the-challenges-counties-face-in-revenue-collection-and-is-automation-of-revenue-an-end-in-itself/

International Monetary Fund. (2018, October 23). *IMF Executive Board Concludes 2018 Article IV Consultation and Establishes Performance Criteria for the Second Review Under the Stand-By Arrangement with Kenya*. Retrieved from imf.org: https://www.imf.org/en/News/Articles/2018/10/23/pr18393-kenya-imf-executive-board-concludes-2018-article-iv-consultation

Jenkins, G. P., Kuo, C.-Y., & Shukla, G. P. (2000). *Tax Analysis and Revenue Forecasting - Issues and Techniques.* Harvard Institute for International Development.

JMP Statistical Discovery. (2023, April 13). *Fitting the Multiple Linear Regression Model*. Retrieved from JMP Statistics Knowledge Portal: https://www.jmp.com/en_hk/statistics-knowledge-portal/what-is-multiple-regression/fitting-multiple-regression-model.html

Johnston, F. R., Boyland, J. E., Meadows, M., & Shale, E. (1999). Some properties of a simple moving average when applied to forecasting a time series. *Journal of the Operational Research Society, 50*, 1267-1271. doi:https://doi.org/10.1057/palgrave.jors.2600823

Kaloev, M., & Krastev, G. (2021). Comparative Analysis of Activation Functions Used in the Hidden Layers of Deep Neural Networks. *3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)* (pp. 1-5). Ankara, Turkey: IEEE. doi:10.1109/HORA52670.2021.9461312

Kothari, C. R. (2004). *Research Methodology: Methods and Techniques* (2nd ed.). New Delhi: New Age International (P) Ltd. Publishers.

Lawson, A. (2015). *Public Financial Management.* Birmingham, UK: GSDRC Professional Development Reading Pack no. 6. Retrieved June 2023, from http://gsdrc.org/docs/open/reading-packs/pfm_rp.pdf

Li, Y., & Ying, F. (2011). Multivariate Time Series Analysis in Corporate Decision-Making Application. *International Conference of Information Technology, Computer Engineering and Management Sciences* (pp. 374-376). Nanjing, China: IEEE.

Lones, M. A. (2024, August 29). *How to avoid machine learning pitfalls: a guide for academic researchers.* doi:https://doi.org/10.48550/arXiv.2108.02497

Lu, L., & Kim, D.-K. (2011). Required Behavior of Sequence Diagrams: Semantics and Refinement. *Proceedings of the 2011 16th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS '11)* (pp. 127–136). USA: IEEE Computer Society.

Madhu, B., Rahman, M. A., Mukherjee, A., Islam, M. Z., Roy, R., & Ali, L. E. (2021). A Comparative Study of Support Vector Machine and Artificial Neural Network for Option Price Prediction. *Journal of Computer and Communications, 9*, 78-91. doi:https://doi.org/10.4236/jcc.2021.95006

Mahmoud, M. A., & Woodall, W. H. (2010). An Evaluation of the Double Exponentially Weighted Moving Average Control Chart. *Communications in Statistics — Simulation and Computation, 39*(5), 933-949. doi:https://doi.org/10.1080/03610911003663907

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, 1346–1364. doi:https://doi.org/10.1016/j.ijforecast.2021.11.013

Mall, R. (2018). *Fundamentals of Software Engineering* (5th ed.). Delhi: PHI Learning.

Maulud, D. H., & Abdulazeez, A. M. (2020). A Review on Linear Regression Comprehensive in Machine Learning. *Journal of Applied Science and Technology Trends, 01*(04), 140-147.

McCurdy, C., Galper, H., & Raza, R. (2019, November 07). *Kenya's Optimistic Revenue Forecasts Are Causing Problems, but Better Tools Can Help*. Retrieved from Urban Institute: https://www.urban.org/urban-wire/kenyas-optimistic-revenue-forecasts-are-causing-problems-better-tools-can-help

Mitchell, T. M. (1997). *Machine Learning.* New York: McGraw-Hill.

Montero-Manso, P., & Hyndman, R. J. (2020). *Principles and Algorithms for Forecasting Groups of Time Series: Locality and Globality.* Monash University, Department of Econometrics and Business Statistics. Clayton, Melbourne, Victoria, Australia: Monash Business School. Retrieved from http://business.monash.edu/econometrics-and-business-statistics/research/publications

Mutua, J., & Wamalwa, N. (2017). *Enhancing Mobilization of Own Source Revenue in Nairobi City County: Issues and Opportunities.* Nairobi: Institute of Economic Affairs.

Mwangi, C. N. (2016). Artificial neural network model for inflation forecasting in Kenya. Nairobi, Kenya. Retrieved from https://su-plus.strathmore.edu/handle/11071/4828

Mwenda, A. K. (2010). *Economic and Administrative Implications of the Devolution Framework Established by the Constitution of Kenya.* Nairobi: Institute of Economic Affairs.

Myers, G. (2015). A World-Class City-Region? Envisioning the Nairobi of 2030. *American Behavioral Scientist, 59*(3), 328-346.

Nairobi City County Government. (2022). *2021/22 FY Quarter 4 Budget Implementation Report.* Finance and Economic Planning. Nairobi: NCCG.

Nairobi City County Government. (2023). *Program Based and Itemised Budget.* Finance and Economic Planning. Nairobi: NCCG.

Office of the Controller of Budget. (2022). *County Governments Annual Budget Implementation Review Report for the FY 21-22.* Nairobi: OCOB.

Onu, F. U., & Ezeji, M. O. (2017). Significance and Application of Computer-Based Forecasting to Governance and Leadership. *International Journal of Computer Applications Technology and Research, 6*(4), 199-208.

Oso, W. Y., & Onen, D. (2009). *A General Guide to Writing Research Proposal and Report* (Revised ed.). Nairobi: The Jomo Kenyatta Foundation.

Rabanser, S., Januschowski, T., Flunkert, V., Salinas, D., & Gasthaus, J. (2020, May 20). *The Effectiveness of Discretization in Forecasting: An Empirical Study on Neural Time Series Models.* doi:https://doi.org/10.48550/arXiv.2005.10111

Raschka, S. (2018, November). Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning. Madison, Wisconsin, United States of America.

Rasul, K., Ashok, A., Williams, A. R., Ghonia, H., Bhagwatkar, R., Khorasani, A., . . . Rish, I. (2024, February 08). *Lag-Llama: Towards Foundation Models for Probabilistic Time Series Forecasting.* doi:https://doi.org/10.48550/arXiv.2310.08278

Ray, S. (2019). A Quick Review of Machine Learning Algorithms. *International Conference on Machine Learning, Big Data, Cloud and Parallel Computing* (pp. 35-39). India: IEEE.

Reddick, C. G. (2004). Assessing Local Government Revenue Forecasting Techniques. *International Journal of Public Administration, 27*(8-9), 597-613. doi:https://doi.org/10.1081/PAD-120030257

Rodrigo, J. A., & Ortiz, J. E. (2024, 11 30). skforecast (Version 0.14.0) [Computer software]. doi:https://doi.org/10.5281/zenodo.8382788

Rubin, M., Mantell, N., & Pagano, M. A. (1999). Approaches to Revenue Forecasting by State and Local Governments. *Proceedings. Annual Conference on Taxation and Minutes of*

the Annual Meeting of the National Tax Association (pp. pp. 205-221). National Tax Association.

Rudakova, O. (2022, November 3). *Deploying AI/ML: It Is a Marathon, Not a Sprint!* Retrieved from FP&A Trends: https://fpa-trends.com/article/deploying-aiml-it-marathon-not-sprint

Salinas, D., Flunkert, V., & Gasthaus, J. (2019, February 22). *DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks.* doi:https://doi.org/10.48550/arXiv.1704.04110

SAS. (2023, June 22). *Predictive Analytics: What it is and why it matters*. Retrieved from SAS: https://www.sas.com/en_us/insights/analytics/predictive-analytics.html

Shamsuddin, S. M., Sallehuddin, R., & Yusof, N. M. (2008). Artificial Neural Network Time Series Modeling For Revenue Forecasting. *Chiang Mai J. Sci, 35*(3), 411-426.

Sharda, R., Delen, D., & Turban, E. (2014). *Business Intelligence and Analytics: Systems for Decision Support* (10 ed.). Pearson Education.

Sherwood, M. (2015). *Medium-Term Budgetary Frameworks in the EU Member States.* European Commission, Directorate General Economic and Financial Affairs (DG ECFIN). Luxembourg: Publications Office of the European Union.

StudySmarter. (2023, May 04). *Quality Criteria: Examples and Standards*. Retrieved from StudySmarter: https://www.studysmarter.us/explanations/psychology/research-methods-in-psychology/quality-criteria/

Swanson, C. J. (2008). Long-Term Financial Forecasting for Local Governments. *Government Finance Review, 24*(5), 60-66.

Talluri, K. T., & Ryzin, G. J. (2014). An Introduction to Revenue Management. In *INFORMS TutORials in Operations Research* (pp. 142-194). Maryland, USA: Institute for Operations Research and the Management Sciences (INFORMS).

Tilley, S., & Rosenblatt, H. (2016). *Systems Analysis and Design* (11th ed.). Boston: Cengage Learning.

United Nations. (2018, January 30). *Local Government SDG Indicator Metadata.* Retrieved from unstats.un.org: https://unstats.un.org/sdgs/metadata/files/Metadata-05-05-01b.pdf

Williams, D. W., & Kavanagh, S. C. (2016). Local Government Revenue Forecasting Methods: Competition and Comparison. *Journal of Public Budgeting, Accounting & Financial Management, 28*(4), 488-526.

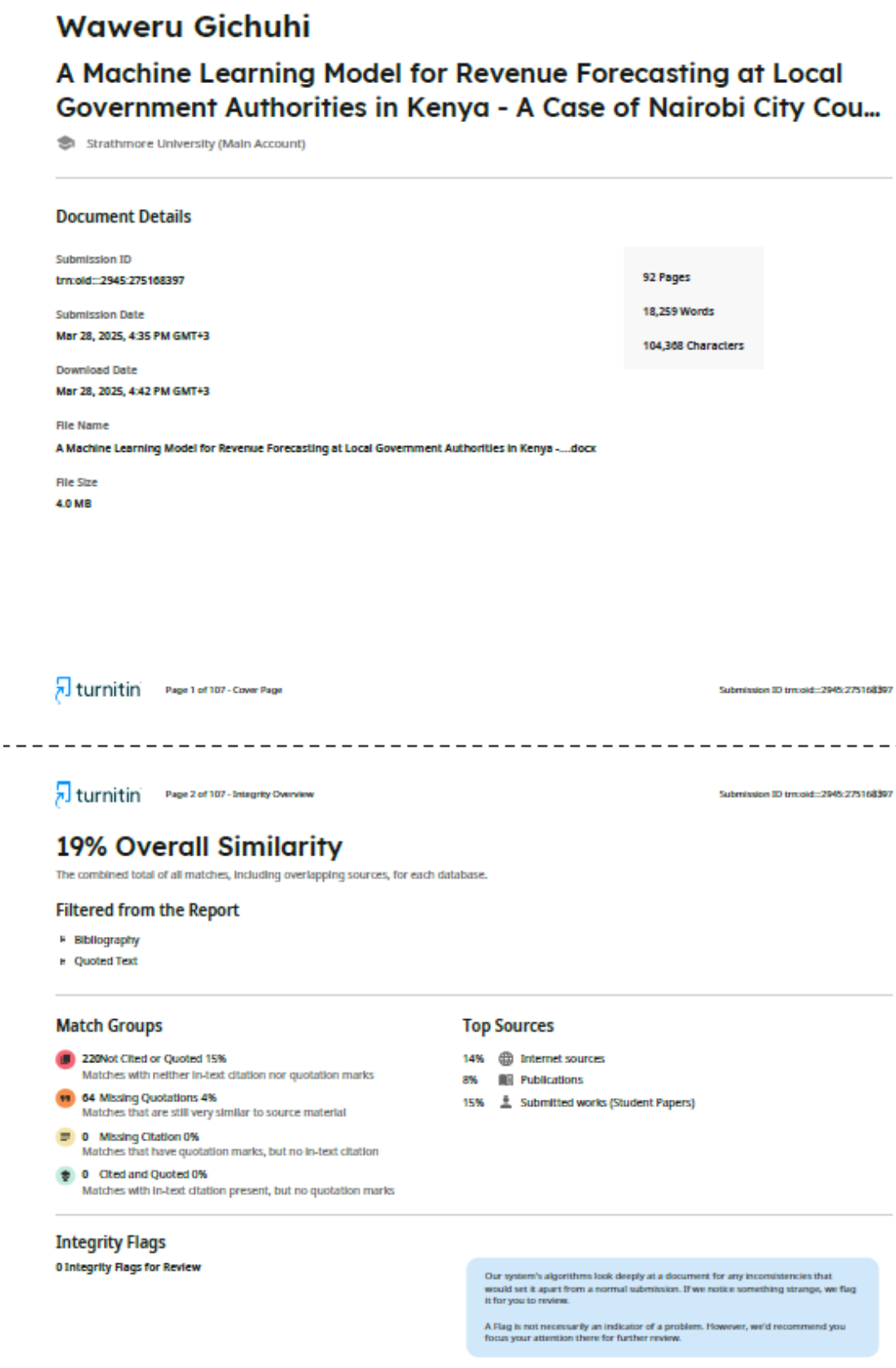
Yang, H., He, J., & Jiang, F. (2018). Hybrid Genetic Algorithm and Support Vector Machine Performance in Public Fiscal Revenue Prediction. *Proceedings of the 37th Chinese Control Conference* (pp. 4495-4499). Wuhan, China: IEEE.

Yingjie, Z., & Abolghasemi, M. (2024, November 10). *Local vs. Global Models for Hierarchical Forecasting.* doi:https://doi.org/10.48550/arXiv.2411.06394

Zhu, L., & Laptev, N. (2017, September 06). Deep and Confident Prediction for Time Series at Uber. San Francisco, California 94103, USA.

Appendices

Appendix A: Similarity Report



Waweru Gichuhi

A Machine Learning Model for Revenue Forecasting at Local Government Authorities in Kenya - A Case of Nairobi City Cou...

Strathmore University (Main Account)

Document Details

Submission ID
trn:oid:::2945:275168397

Submission Date
Mar 28, 2025, 4:35 PM GMT+3

Download Date
Mar 28, 2025, 4:42 PM GMT+3

File Name
A Machine Learning Model for Revenue Forecasting at Local Government Authorities in Kenya -....docx

File Size
4.0 MB

92 Pages

18,259 Words

104,368 Characters

turnitin Page 1 of 107 - Cover Page Submission ID trn:oid:::2945:275168397

turnitin Page 2 of 107 - Integrity Overview Submission ID trn:oid:::2945:275168397

19% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

220 Not Cited or Quoted 15%
Matches with neither in-text citation nor quotation marks

64 Missing Quotations 4%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

14% Internet sources
8% Publications
15% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Appendix B: Ethical Clearance Confirmation

Strathmore UNIVERSITY

26th June 2023

Mr Gichuhi Waweru,
waweru.gichuhi@strathmore.edu

Dear Mr Gichuhi,

RE: A Machine Learning Model for Revenue Forecasting at Local Government Authorities in Kenya: A Case of Nairobi City County

This is to inform you that SU-ISERC has reviewed and **approved** your above SU-masters research proposal. Your application reference number is **SU-ISERC1776/23.** The approval period is from **26th June 2023 to 25th June 2024.**

This approval is subject to compliance with the following requirements:

i. Only approved documents including (informed consents, study instruments, MTA) will be used.
ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-ISERC.
iii. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-ISERC within 72 hours of notification.
iv. Any changes anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-ISERC within 72 hours.
v. Clearance for the export of biological specimens must be obtained from relevant institutions.
vi. Submission of a request for renewal of approval at least 60 days prior to the expiry of the approval period. Attach a comprehensive progress report to support the renewal.
vii. Submission of an executive summary report within 90 days of completion of the study to SU-ISERC.

Before commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology, and Innovation (NACOSTI) https://research-portal.nacosti.go.ke/ and obtain other clearances needed.

Yours sincerely,

for: **Mr Ambrose Rachier,**
Chairperson; SU-ISERC

STRATHMORE UNIVERSITY INSTITUTIONAL SCIENTIFIC AND ETHICAL REVIEW COMMITTEE
(SU-ISERC)
26-Jun-2023
Email:ethicsreview@strathmore.edu
P.O BOX 59857-00200
NAIROBI-KENYA

Appendix C: NACOSTI Permit

REPUBLIC OF KENYA

NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Ref No: 249764

Date of Issue: 17/July/2023

RESEARCH LICENSE

This is to Certify that Mr.. Waweru Gichuhi of Strathmore University, has been licensed to conduct research as per the provision of the Science, Technology and Innovation Act, 2013 (Rev.2014) in Nairobi on the topic: A Machine Learning Model for Revenue Forecasting at Local Government Authorities in Kenya: A Case of Nairobi City County for the period ending : 17/July/2024.

License No: NACOSTI/P/23/27538

249764

Applicant Identification Number

Director General
NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Verification QR Code

NOTE: This is a computer generated License. To verify the authenticity of this document, Scan the QR Code using QR scanner application.

See overleaf for conditions

Appendix D: Data Collection Tool

NCCG Own Source Revenue Collection and Forecasting Research

1 response

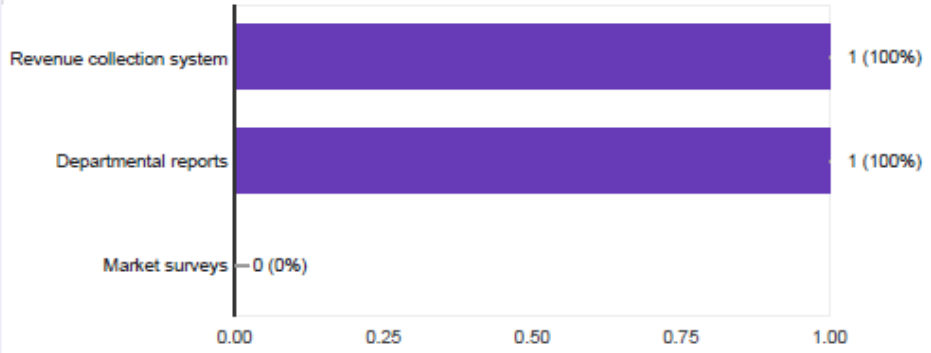
Which department(s) is/are responsible for generating own source revenue (OSR) forecasts at Nairobi City County?

1 response

Revenue

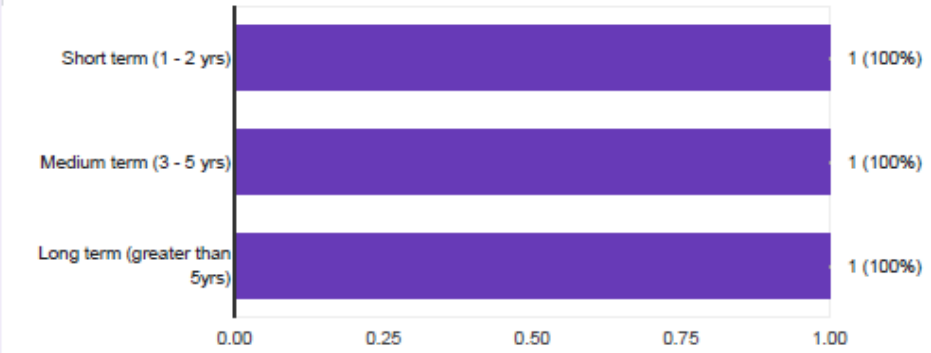
Which sources of data are used in the OSR forecasting process? (please tick all that apply) Copy

1 response



What is the outlook period for which OSR forecasts are projected? (please tick all that apply) Copy

1 response



Are there baseline formulae used to generate forecasts for the following types of revenue sources? (please tick below if response is "yes", leave blank if "no")

0 responses

No responses yet for this question.

For revenue sources that have a forecasting formula, please list them below, then use the attached spreadsheet in Section 2 to provide sample methods

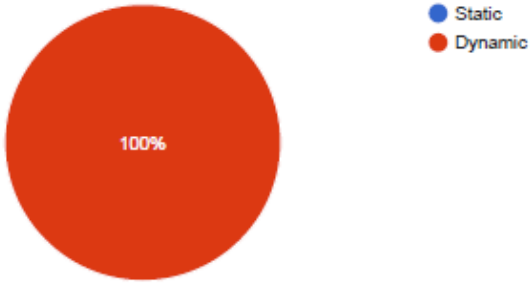
0 responses

No responses yet for this question.

Once OSR forecasts are submitted and approved, do they remain static for the duration of the reporting period or do they change based on dynamic socio-economic conditions?

Copy

1 response



Which metrics are used to gauge revenue collection performance after forecasting is done?

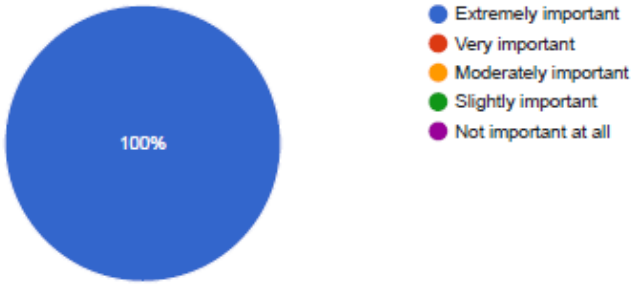
1 response

Actual vs perfomance

How crucial is expert judgement by line departments in determining OSR forecasts?

Copy

1 response



What are some of the key challenges faced when generating OSR forecasts?

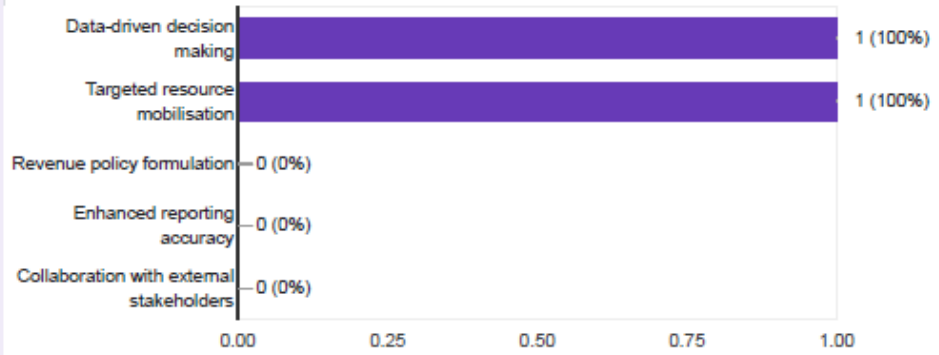
1 response

Data availability especially for unstructured revenues

How would system-generated forecasts help the county achieve its goals? (please tick all that apply)

Copy

1 response



Data Collection Template

Link to spreadsheet

0 responses

No responses yet for this question.

Appendix E: Dataset

Data Dictionary

- **DateStamp:** A month-start date representing a calendar month
- **FiscalYear:** The financial reporting period used by County Governments in Kenya
- **FiscalQuarter:** Three-month ordinal periods starting on 01st July and ending on 30th June in each fiscal year
- **RevenueType:** Categories assigned to the major sources of revenue
- **TransactionCount:** The number of individual payments that make up revenue collected in a given month
- **MonthlyCollection:** The monthly revenue collected per stream in Kenya Shillings

	FiscalYear	FiscalQuarter	RevenueType	TransactionCount	MonthlyCollection
count	840	840	840	802.00	802.00
unique	5	4	14	NaN	NaN
top	2018/2019	Q1	Billboards & Advertisement	NaN	NaN
freq	168	210	60	NaN	NaN
mean	NaN	NaN	NaN	5693.70	48067759.38
std	NaN	NaN	NaN	20221.52	188716929.36
min	NaN	NaN	NaN	0.00	0.00
25%	NaN	NaN	NaN	5.00	76125.00
50%	NaN	NaN	NaN	103.50	3416170.00
75%	NaN	NaN	NaN	1012.25	32330224.75
max	NaN	NaN	NaN	177176.00	3444340243.00

	MonthlyCollection							
	count	mean	std	min	25%	50%	75%	max
RevenueType								
Billboards & Advertisement	60.00	46620219.13	36217827.38	0.00	22326975.00	39745516.50	69802881.50	144051174.00
Decentralisation	57.00	1148844.16	1068568.67	0.00	275261.00	912741.00	1774927.00	4794470.00
Education Fees	52.00	22607.94	22375.54	0.00	5000.00	14500.00	42250.00	80000.00
House and Stall Rent	49.00	15850433.29	23933650.58	0.00	850.00	4738415.00	17684078.00	87186452.00
Land Rates	60.00	35766782.37	100247228.97	0.00	1327932.50	2050522.50	7155239.50	568264599.30
Market Fees	60.00	17357036.43	14205568.25	0.00	4015902.00	15351099.00	26559368.25	58556112.00
Mortuary Fees	59.00	2641345.29	2266880.65	0.00	435550.00	2779401.00	3676977.50	12376358.00
Other Income	60.00	164677961.20	567873010.92	0.00	21282548.75	34342414.50	71511995.50	3444340243.00
Parking Fees	60.00	47953841.27	73976295.39	0.00	2335587.25	5872910.50	63051062.50	280125062.10
Plan Inspection and Approvals	60.00	46215427.74	37398150.97	0.00	10098815.00	41774871.00	70662028.25	130215726.00
Rent	45.00	136122.64	403564.14	0.00	600.00	42400.00	166100.00	2700000.00
SDR	60.00	4991024.97	5628834.64	0.00	450635.00	4261745.00	6200183.75	27346539.00
Single Business Permits	60.00	260321645.92	261576032.37	0.00	86919003.25	183471578.50	331547380.00	1052167641.00
TPS	60.00	1846847.10	2325327.97	0.00	433388.25	940209.50	2570224.25	11523699.00

Appendix D: Code Snippet

```
# Perform a Seasonal-Trend decomposition using LOESS (STL) to extract seasonality, trend and residual noise
for rev in df["RevenueType"].unique():
    stl_rev = STL(df[df.RevenueType==rev].MonthlyCollection, # Fit local regressions for every revenue type
                  period=12, # Yearly reccurence in revenue collection patterns
                  seasonal=13, # Smooth over the annual cycle
                  robust=True).fit() # Method is robust to outliers

    # Store category-wise residual values in a new column
    df["Residual"] = stl_rev.resid

    # Compute the inter-quartile range for each revenue type
    df["Q1"] = df[df.RevenueType==rev].Residual.quantile(0.25)
    df["Q3"] = df[df.RevenueType==rev].Residual.quantile(0.75)
    df["IQR"] = df["Q3"] - df["Q1"]

    # Test using 1.5 x IQR rule (covers 99.7% of a normal distribution)
    factor = 1.5
    df["LowerLimit"] = df["Q1"] - factor * df["IQR"]
    df["UpperLimit"] = df["Q3"] + factor * df["IQR"]

    # Flag as "true" if residual value lies beyond these limits
    df["IsOutlier"] = (df.Residual < df.LowerLimit) | (df.Residual > df.UpperLimit)

    # Finally, plot the decomposition components
    fig, axs = plt.subplots(4, 1, figsize=(15,17))

    axs[0].plot(stl_rev.observed)
    axs[0].set_title(str(rev) + " Revenue")

    axs[1].plot(stl_rev.trend)
    axs[1].set_title(str(rev) + " Trend")

    axs[2].plot(stl_rev.seasonal)
    axs[2].set_title(str(rev) + " Seasonality")

    axs[3].plot(stl_rev.resid, linestyle="none", marker="o")
    axs[3].set_title(str(rev) + " Residuals & Outliers")

    # If any data points are identified as outliers, plot them
    if df["IsOutlier"].any():
        df.Residual.loc[df["IsOutlier"]].plot(marker="o", color="r", linestyle="none")
```