University of Kerbala
College of Computer Science & Information Technology
Computer Science Department

FOOD SALES PREDICTION USING MACHINE LEARNING TECHNIQUES

A Thesis

Submitted to the Council of the College of Computer Science & Information Technology / University of Kerbala in Partial Fulfillment of the Requirements for the Master Degree in Computer Science

Written by

Hussam Mezher Merdas Bresam

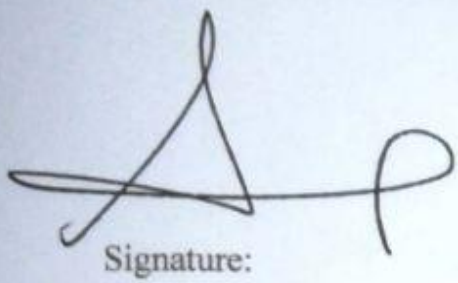
Supervised by

Assist. Prof. Dr. Ayad Hameed Mousa

2023 A.D. 1444 A.H.

Supervisor Certification

I certify that the thesis entitled (**Food Sales Prediction Using Machine Learning Techniques**) was prepared under my supervision at the department of Computer Science/College of Computer Science & Information Technology/ University of Kerbala as partial fulfillment of the requirements of the degree of Master in Computer Science.



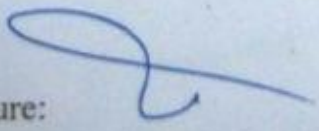
Signature:

Supervisor Name: Assist. Prof. Dr. Ayad Hameed Mousa

Date: ١٢ / ١٢ /2023

The Head of the Department Certification

In view of the available recommendations, I forward the thesis entitled "Food Sales Prediction Model Using Machine Learning Techniques" for debate by the examination committee.



Signature:

Assist. Prof. Dr. Muhannad Kamil Abdulhameed

Head of Computer Science Department

Date: / /2023

Certification of the Examination Committee

We hereby certify that we have studied the dissertation entitled (**Food Sales Prediction Using Machine Learning Techniques**) presented by the student (**Hussam Mezher Merdas**) and examined him/her in its content and what is related to it, and that, in our opinion, it is adequate with (**Very Good**) standing as a thesis for the degree of Master in Computer Science.

Signature:
Name: Dr. Asia Mahdi Naser
Title: Prof.
Date: / 12 / 2023
(Chairman)

Signature:
Name: Dr. Noor Dhia Kadhem
Title: Assist. Prof.
Date: / 12 / 2023
(Member)

Signature:
Name: Dr.Wathiq Laftah Abd-Ali
Title: Assist. Prof.
Date: / 12 / 2023
(Member)

Signature:
Name: Dr. Ayad Hameed Mousa
Title: Assist. Prof.
Date: 13 / 12 / 2023
(Member and Supervisor)

Approved by the Dean of the College of Computer Science & Information Technology, University of Kerbala.

Signature:
Assist. Prof. Dr. Mowafak Khadom Mohsen
Date: / / 2023
(Dean of Collage of Computer Science & Information Technology)

Dedication

To those who supported me at this important stage, Assist. Prof. Dr. (Ayad Hameed Mousa). To those who are always by my side, my dear parents, my wife, my son, my family, my friends, and everyone who supported me. Thank you very much.

Acknowledgement

My sincere thanks to ALLAH Almighty for all the blessings He has bestowed upon me and for the knowledge He has provided me with. I would also like to thank my supervisor, Assist. Prof. Dr. (Ayad Hameed Mousa), for his continuous support. I also thank my dear parents for their continuous efforts with me. I also extend my thanks to my dear companions: my wife, and my beloved son. Thanks to my family, friends, and everyone who supported me.

Hussam Mezher Merdas

2023

Abstract

Nowadays, the economy of nations is based on the development of the private sector. One of the most important companies working in this sector is food companies. These companies have suffered from the problem of storage and damage to products due to their expiration date. This thesis proposes building a strategy that predicts the actual need of companies for quantities of products in the presence of other variables. The proposed strategy is based on two main pillars: The first is using three different food datasets with a different correlation between their features, as the first dataset is of high correlation, the second is of a medium correlation, and the third will be a weak correlation. The second is using thirteen Machine Learning algorithms and evaluating their results based on several specific metrics to obtain the best algorithm in terms of accuracy. The obtained results indicate that the best algorithm applied to the first dataset with a high correlation is Gradient Boosting which gave an accuracy (98.65), and the best algorithm applied to the second dataset with a medium correlation is also Gradient Boosting which gave an accuracy (59.29), while the best algorithm applied to the third dataset with a low correlation was Random Forest which gave accuracy (39.39). Based on these results, the proposed model was applied to predict the quantities of Dates that Iraq will produce for the next five years, with the availability of other variables. Where the dataset was of good correlation, the first algorithm was used and gave accuracy (99.51). This thesis sets a limit to the daily food wastage that incurs significant financial losses worldwide.

Declaration Associated with this Thesis

Some of the works presented in this thesis have been published or accepted as listed below.

- The research article titled “Food Sales Prediction Using MLP, RANSAC, and Bagging” has been accepted by “Journal of Techniques”.
- The research article titled “Utilizing Ridge Techniques to Combat Food Waste” has been published in “Journal of Engineering Science and Technology Review”.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Abbreviation	Description
AI	Artificial Intelligence
ANFIS	Adaptive Network-based Fuzzy Inference System
ANN	Artificial Neural Network
ARIMA	Autoregressive Integrated Moving Average
ATLAS	Advanced Temporary Latent Factor Approach to Sales Forecasting
DL	Deep Learning
GBR	Gradient Boosting Regression
GPR	Gaussian Processes Regression
KNN	K-Nearest Neighbors
KRR	Kernel Ridge Regression
LASSO	Least Absolute Shrinkage and Selection Operator
LGBM	Light Gradient Boosting Machine Learning
LSTM	Long-Short Term Memory
MAE	Mean Absolute Error
ML	Machine Learning
MLP	Multilayer Perceptron
MSE	Mean Squared Error
RANSAC	Random Sample Consensus
RBF	Radial Basis Function
RF	Random Forest
RMSE	Root Mean Squared Error

RMSLE	Root Mean Squared Logarithmic Error
RNN	Recurrent Neural Network
SMAPE	Symmetric Mean Absolute Percent Error
SVM	Support Vector Machine
SVR	Support Vector Regression
TFT	Temporal Fusion Transformer
XGBoost	Extrema Gradient Boosting

CHAPTER ONE

INTRODUCTION

1.1 Overview

Many food sales companies relied on very primitive methods in estimating the quantities that they would need to sell in the market. The managers of these companies employed many employees with specializations in mathematics and economics to increase the profits of their companies by reducing the losses resulting from goods damage, as a result of the expiration of their storage [1]. Many companies are competing with each other. Each of these companies wants to increase its profits more than its competitors [2]. On the other hand, they desire the least losses in terms of employee salaries and numbers, as well as in terms of reducing product losses as a result of their damage due to their remaining in the companies' warehouses without selling them because the availability of these goods is more than the demand for them [3].

From this standpoint, and to achieve the best profits, companies have developed research and scientific aspects to reduce the burden of product loss [4]. The researchers took from the past sciences as an important pillar and proceeded to use modern technologies to develop efficient forecasting systems to estimate the future need for food sales. This has been of great benefit to companies by storing only the quantities they will need without falling into the problem of stockpiling goods [5].

When taking a look at the development of modern prediction systems; It can start from the period of the emergence of Artificial Intelligence (AI), and here we mean specifically Machine Learning (ML) algorithms in the areas of classification and regression. The concern of this study is the prediction algorithms, as there are many of these algorithms and they vary in accuracy according to the type of this algorithm as well as the dataset used. Then, the development of AI

algorithms showed deep learning algorithms, which also gave good results. Many studies have also been conducted to develop this important scientific field, as well as to obtain efficient results in terms of accuracy.

By reviewing and analyzing the relevant studies, it was found that they differed in terms of choosing the data on which they depend, depending on the companies that provide this data. On the same aspect, food datasets often need a lot of pre-processing before they can be used for prediction. After the stage of preparing the datasets, the stage of training the algorithms begins and then testing them to see their accuracy in prediction to use them later, whether in future studies or for the benefit of the food-selling companies' sector.

1.2 Problem Statement

The main goal of establishing companies is profit. And the most important thing that hinders and reduces the profit of companies selling food products is the spoilage and damage of foodstuffs. One of the most important reasons for food spoilage is the expiration date, causing significant financial losses for individuals, communities, and companies. The cause of these losses is the inability of companies to accurately anticipate the actual demand for these products. Previous research was limited to the use of a few food datasets and algorithms applied to them. These studies gave results of varying accuracy. Many companies do not want to waste more time and money to create appropriate prediction models for their perishable foodstuffs [6].

1.3 Research Objectives

The main aim of this study is to design and implement a food sales prediction models that achieves multiple objectives:

1. Predict the actual quantity of products that companies need in the future to avoid spoilage caused by expiration.

2. Use three different datasets in terms of data type and the correlation of their features to find out the extent of their actual future impact on the prediction results. And applying thirteen various ML algorithms to the relevant datasets after making pre-processing for datasets. Then, using different metrics, an efficient comparison is made between the accuracy obtained from each algorithm to find out the feasibility of predicting algorithms and how the correlation of dataset features affects the results obtained.

3. Use an actual local dataset of Dates sales in Iraq for the period (2002-2020) and then predict the quantities of Dates that Iraq will produce for the next five years.

1.4 Thesis Organization

This thesis contains five chapters. A short background that tops each of these chapters, clarifying his contributions and giving an impression of what the results achieved in this chapter will lead to. We can summarize each chapter as follows:

Chapter 1: This chapter provides the reader with an important objective introduction that discusses the main points on which the thesis is based. It also gives a summary of the problem statement. It also briefly presented the most important goals and aspirations of this study. The introduction

also covers urgently the most important contributions that this study will add to the scientific community and the practical and economic side of the country.

Chapter 2: This chapter focuses on the theoretical side of the thesis. It takes the reader to the merits of AI from the moment of its inception to its current advanced stages and how it was reflected in the field of prediction. This chapter also looks at the most important previous studies, how datasets were used, how AI algorithms were employed, and the accuracy obtained from each proposed model of these studies.

Chapter 3: This chapter forms the heart of the study. Through which the proposed system can be known. Where will cover through it the necessary steps to establish the proposed system. This chapter will include the entire practical aspect, from the stage of selecting datasets to the stage of pre-processing them, up to the stage of applying ML algorithms. The reader will finally be able to compare the results obtained and feel with his hands the actual accuracy achieved for each AI technology used.

Chapter 4: The fourth chapter lists the results achieved from the proposed system, focusing on two main aspects, the first is the type of dataset used, as well as the accuracy of the ML algorithm concerned. Also, in this chapter, testing the proposed model using a real dataset will be discussed.

Chapter 5: This chapter deals with the most important practical and scientific benefits offered by the study to be relied upon by subsequent future studies. The chapter also provides an objective conclusion of all that was previously presented in the thesis.

CHAPTER TWO

THEORETICAL BACKGROUND

2.1 Overview

Machine learning (ML) is an important and fundamental branch of AI, which is defined as the ability of a machine to accurately simulate human behavior [7]. AI systems try to perform the functions entrusted to humans by performing the tasks that humans perform and sometimes even tasks that they cannot easily perform.

This study relies on supervised ML. Specifically, the proposed model is based on the regression technique. Where the AI algorithms used are employed on real and continuous data to give forecast results for food sales, then the accuracy obtained from these algorithms is evaluated by comparing the results obtained from them with the actual numbers in the datasets.

2.2 The Correlation between Features of Datasets

It is important to understand the relationship between the features in the dataset because it has an impact on the prediction results. Correlation can be defined as the effect of one variable on another variable, positively or negatively. Since the correlation can be positive or negative, that is, the two variables change in one direction, on the other hand, the correlation can be negative between the features, so that one of them changes opposite to the other, or what is called negative or reverse change. It is also possible for the correlation to be zero, that is, neutral so that one variable does not affect another variable [8].

If there is a feature that affects another feature and they are closely related to each other, this may negatively affect the results of the

algorithm, and to achieve good results, this correlation should be disengaged or the variable that negatively affects the result should be deleted. It is worth noting that attention must be paid to the input variables and their relationship to the output variable to select the inputs that positively affect the result to obtain more accuracy [8].

In this thesis, the Pearson correlation coefficient was used to measure the correlation between features in the dataset. The Pearson correlation coefficient, often denoted as (r), is a measure of the linear relationship between two variables. It ranges from -1 to 1. The Pearson correlation coefficient is represented mathematically [9] :

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad (2.1)$$

Where:

n: is a number of values.

x_i , y_i : variabels.

2.3 Models Applied

To make a comprehensive and efficient comparison, thirteen algorithms are tested including ML algorithms, as well as models that rely on ensemble techniques (techniques that are based on merging several models instead of using a single model to obtain greater prediction accuracy). These algorithms are the most important and widespread in the field of forecasting food sales. These algorithms are: (Multilayer Perceptron, Support Vector Regressor (SVR), Multiple Linear Regression, Decision Tree, Random Forest Regressor, K-Nearest

Neighbors (KNN), Bagging Regressor, Gaussian Processes Regressor, RANSAC, Gradient Boosting, Elastic Net, Bayesian Ridge, and Kernel Ridge). Below will discuss the theoretical side of these algorithms in a way that benefits the reader and concerns this study.

2.3.1 Multilayer Perceptron (MLP)

Deep learning is based on the use of multi-layered Artificial Neural Networks to train them. In the fifties of the last century, the Rosenblatt perceptron model was developed[10] , but the topic of neural networks did not receive this important interest until 1986, in this year Dr. Hinton and his colleagues developed the backpropagation technique for training a multi-layered neural network [11]. Now the topic of neural networks has become a popular field for large companies, why it gives good results in prediction.

When the neural network has multiple layers and these layers are fully connected then it is called Multilayer Perceptron (MLP). The simplest MLP networks consist of only three layers, the input layer, and the output layer, and between them, there is one hidden layer. But if there are many hidden layers, then in this case the neural network is called a deep artificial neural network (ANN). MLP is a good example of a feedforward ANN (Figure 2.1). The number of layers in each neural network must be set. The backpropagation balances the weights. The deeper the neural network, the more accurate it is. However, too many deep layers can lead to problems. The inputs, which are weights, are entered into the activation function. The activation function converts the

weights into output values and converts them to the output of the neuron. The activation function is so named because it controls the threshold for activating the neurons and the strength of the output signal [12].

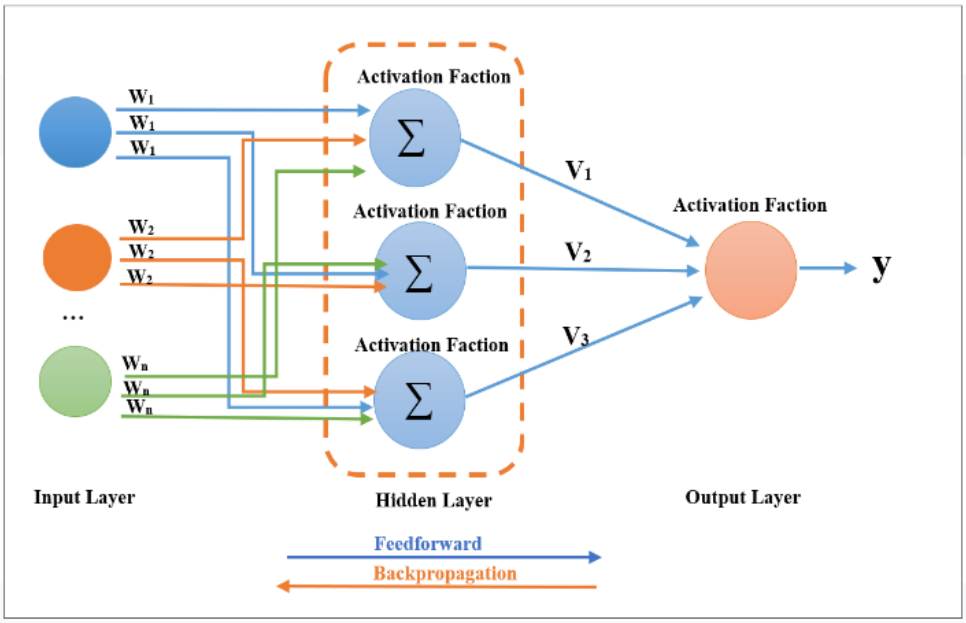


Figure 2.1: Multilayer Perceptron.

2.3.2 Support Vector Regression (SVR)

Support Vector Machine (SVM) is widely used as a classification algorithm. However, this algorithm has good use with regression. It relies on non-linearity in building the model and dealing with the data to produce the prediction [13]. Whereas, when SVM is used for regression purposes, it is called Support Vector Regression or SVR. SVR is a supervised algorithm used for prediction purposes. SVM is a classifier, which performs the classification process by predicting discrete labels while SVR acts as a regression technique used to predict continuous variables [14]. The idea of SVM work is based on the principle of forming

a hyperplane, the same principle is applied in the algorithm SVR, with the difference in the method of determining the best-fit line which represents the hyperplane that contains the maximum number of points [15]. SVM works to reduce the error rate, but SVR works by fitting the error value within a specified threshold, which results in the best possible value within a certain margin [16].

SVM's job is to find the best line(hyperplane) separating two classes. Then it classifies the new point depending on whether it is in the positive first class or the negative second class [17]. Whereas SVR performs the prediction process by focusing on the points within the decision boundary lines. The best prediction line represents the hyperplane with the most points [18].

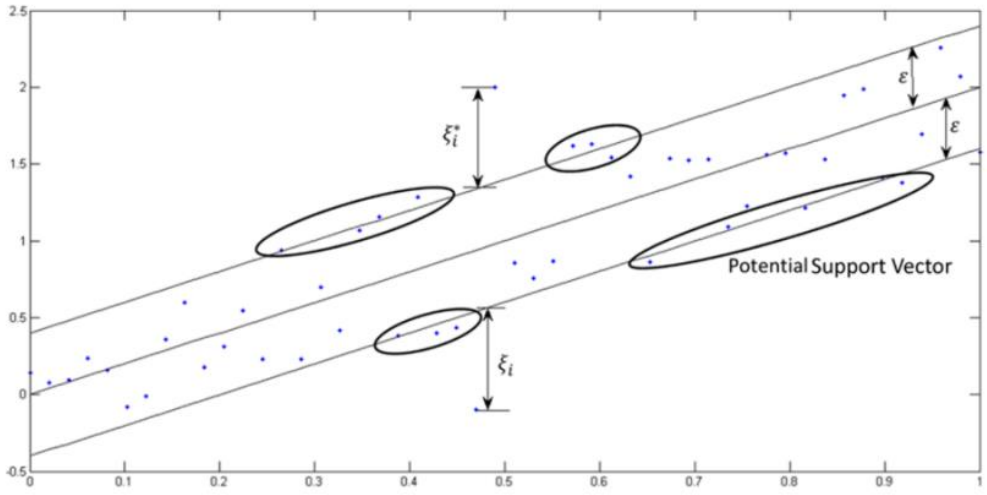


Figure 2.2: Support Vector Regression.[19].

Figure 2.2 above shows that the hyperplane lies in the middle and the boundary lines are two lines on either side of the hyperplane and the distance between them and the hyperplane is the threshold "**Ɛ**". The input data points are dots and "ξ" denotes to soft margin. Data points outside

the boundary are surrounded by an ellipse to indicate that the margin has not been set for those points because they are located within the soft-margin area [20]. The goal of this process is to reduce the error by using a function that puts more original points between the boundary lines while at the same time reducing the "slack "ξ". That gives a more accurate prediction process.

2.3.3 Multiple Linear Regression

Based on one or more variables, multiple linear regression generates a prediction. It is a type of linear regression. This algorithm is based on the prediction of a variable called the dependent variable and builds its prediction on several other variables called the independent variables [21]. Simple linear regression is also a type of linear regression that uses one independent variable to predict another variable [22]. This type of linear regression attempts to create a straight line to represent these two variables.

Multiple Linear regression depends on a linear relationship between two or more variables. When the dependent and independent variables do not follow a single line, then the relationship in this case is non-linear. Can be written multiple linear regression equation as follows [23]:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 \cdot \cdot \cdot + \beta_k X_k + \varepsilon \qquad (2.2)$$

- Y_i is the dependent value.
- β_1 and β_2 are parameters.
- X_i is the independent variable.
- ε is the random error.

2.3.4 Decision Tree

From the shape and structure of trees in nature came the design of the decision tree, which is a well-known supervised learning algorithm that is used in classification and regression. This study focused on regression to form a prediction model. A decision tree consists of three types of nodes. The root node is the main node and the pillar from which the rest of the nodes branch [24]. Dataset features are internal nodes and branches that represent decision rules, and leaf nodes represent the result. Figure 2.3.

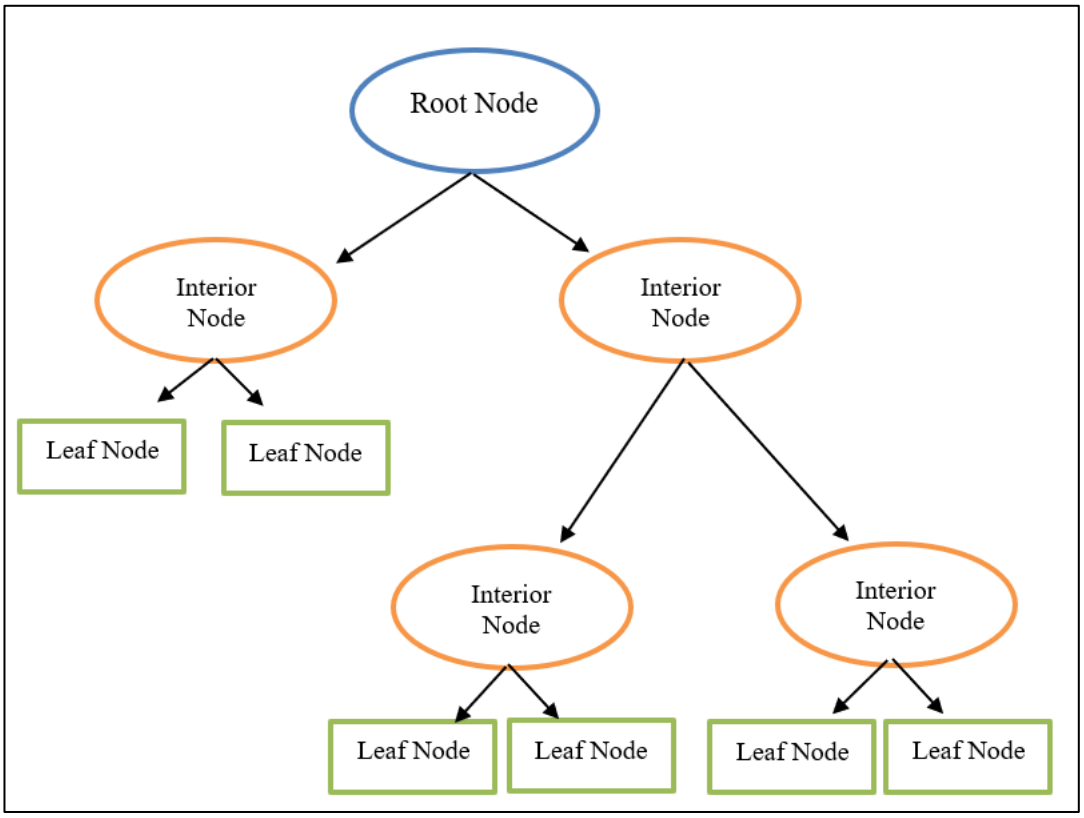


Figure 2.3: Simple Decision Tree.

From the root node, the branching process begins, as it continues for several levels according to the data that is processed until it reaches the

terminal node that contains the prediction, which represents the final result of the algorithm. Branching or division usually starts from the top, as it produces a new variable at each step that divides better. Each subtree is divided into two branches only [24].

In regression, the tree gives leaf nodes with continuous values (usually real numbers), unlike in classification, where the values are discrete [25]. The regression decision tree divides the data into subgroups and fits the model to each subgroup separately and progresses one level after another until it reaches the best possible prediction [26].

2.3.5 Random Forest Regression

Random Forest is one of the supervised learning algorithms based mainly on ensemble learning using several decision trees together [27]. This algorithm adopts the bagging technique, so the processing is done in parallel so that each decision tree works separately [28]. This algorithm can be used in classification and regression, in this study the focus will be on regression. The name of the random forest came from the idea of random bagging of data, and based on several decision trees, a prediction is generated like (forest). Overall, this algorithm is an important and powerful algorithm that minimizes the defects of the decision tree. Moreover, random forest is very popular in the field of sales forecasting for its power and accuracy of results [29].

A random forest is an algorithm that consists of a set of decision trees. These decision trees are generated "randomly" to assemble into a random forest. Decision trees are created by selecting samples from the

rows, and at each point, a division is made based on the features of the datasets [30]. Each decision tree produces its own sub-predictor. Then the final result is produced from the average of the results of the sub-trees. The average yields a random forest with efficient results compared to a single decision tree, which results in high accuracy away from overfitting.

Each decision tree is trained on a data set individually in the formation of the random forest as a whole. At each decision node, the splitting is done based on a random set of features. When the data is complex, non-linear, and continuous, it is preferable to use the random forest algorithm because of its great ability to segment data and deal with large-scale data to obtain ideal results [31].

2.3.6 K-Nearest Neighbors (KNN)

The K-Nearest Neighbors (KNN) algorithm is a non-parametric supervised artificial intelligence algorithm that is easy to implement and simple [32]. Evelyn Fix and Joseph Hodges developed KNN in 1951 [33]. KNN is used in classification and regression. For classification, this algorithm assigns a label to each class using a majority vote, the point's affiliation to any class is determined by adopting the majority vote of its neighbors. The KNN algorithm works in regression as it does in classification, but the difference is that this algorithm uses to predict discrete values in the classification, while it works to predict continuous values in regression, also the output of the prediction represents the value of the object's property. This value is obtained from the average of the k-nearest neighbors [34]. However, before classification is made, distance

or similarity is measured. The best measure of distance is the Euclidean distance [35]. Also, the value of k must be determined based on the entered data. As the value of k affects the prediction results positively or negatively.

As mentioned above, KNN is used in regression to predict continuous values. This algorithm is based on a weighted average of k nearest neighbors. KNN work by determining the value of K as the number of neighbors, then calculating the distance between each point in the data entered with the remaining points using the Euclidean equation or other distance measures. Then the distances are arranged from the largest to the smallest, and then the first K entries are selected from the group that was arranged. Then the labels of the given K entries are obtained. Finally, the mean of the K labels is returned [35].

2.3.7 Bagging Regression

A Bagging Regression is a meta-estimator work based on an ensemble technique suggested by Leo Breiman in 1996 [36]. This algorithm works by taking random samples from the original dataset and then fitting them into regressors after which it aggregates its predictions (either by voting or by averaging) to get a final prediction. This method mainly aims to get rid of overfitting problems in regression and reduce variance within a noisy dataset [37]. A Bagging regressor improves the accuracy and performance of machine learning algorithms. The work of this algorithm can be summarized in three main steps [38]:

1. Bootstrapping: The Bagging relies on bootstrapping sampling technique to generate various samples. This resampling process generates new training samples by randomly selecting and replacing data points. Thus, each time a data point is selected from the training data set.

2. Parallel training: The resulting samples from the first step are trained independently using either basic or weak learners.

3. Aggregation: finally, to form the final estimate the average of all the predictions that resulted from the individual predictors is taken; This process is known as soft voting.

2.3.8 Gaussian Processes Regression

Gaussian Processes Regression (GPR) is a supervised ML algorithm that deals with probabilistic classification and regression problems. This method works well with small datasets as it can provide measures of uncertainty in predictions [39]. GPR differs from the rest of the supervised ML algorithms in that the latter learns the exact values of all parameters in a function, unlike GPR which is based on the principle of probabilities distribution over all possible values [40]. GPR is a non-parametric Bayesian algorithm that works by distributing probabilities over all admissible functions that fit the data and not by calculating the probability of a parameter's distribution on a custom function as shown in figure 2.4 (a and b) [41]. Where the posterior is calculated using the training data, and the predictive posterior distribution is calculated using points of interest.

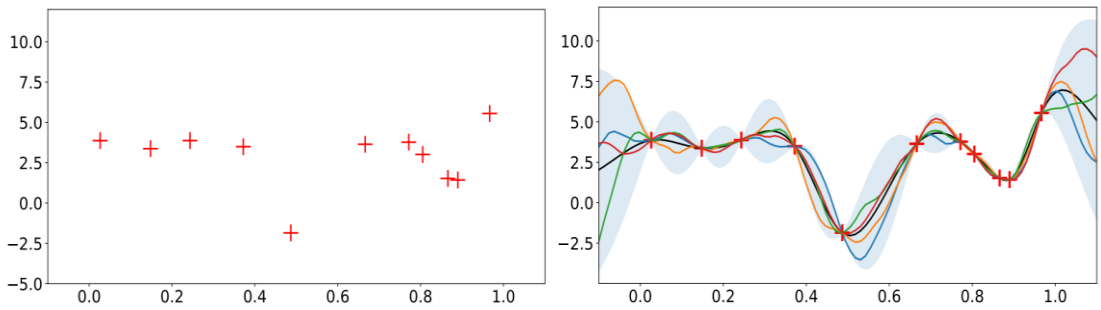


Figure 2.4.a. The observed data points. b. Five possible functions of GPR[41].

For more clarification, suppose that the random variable is denoted by X and x is the real argument, where it is Gaussian or normally distributed depending on the average μ and assuming that the variance is σ2 if the probability density function (PDF) is as follows [42]:

$$P_X(x) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) . \tag{2.3}$$

For the above, the GPR algorithm is given Uncertainty estimates with prediction values. Also, kernel functions are in the process of integrating prior knowledge of the form of functions [43].

2.3.9 Random Sample Consensus (RANSAC)

RANSAC is an iterative ML algorithm, which is an abbreviation for (Random sample consensus) proposed by Fischler and Bolles in 1981 [44]. This algorithm estimates the parameters of a model by taking random samples from the target data. Where the dataset that contains inliers and outliers is used, RANSAC is considered efficient in terms of isolating the required data from the outliers and thus forming an efficient predictive model [45]. RANSAC is used for regression because, as noted, it is good

at handling outliers. This algorithm works by randomly selecting a subset of data samples and then using these samples to estimate the model parameters. Then in the next step, RANSAC finds samples that are within the fault tolerance range of the model. These samples are named inliers data and form a consensus set, and the rest data is named outliers data [46]. This algorithm trains the model using the inliers data and iterates these steps several times to produce a more efficient and less error-prone model.

This algorithm does not guarantee to obtain the best parametric model due to its randomness. However, the possibility of obtaining the optimal model remains possible by assigning appropriate values to the algorithm's parameters [47].

One of the most important features of RANSAC is its ability to perform an efficient estimation of model parameters despite the presence of a large number of outliers in the target data set [48]. And the most important defect of this algorithm is that there is no specific time for calculating these parameters. Whereas, when a limited number of iterations is used, the optimal or desirable solution may not be obtained.

2.3.10 Gradient Boosting Regression

Gradient Boosting Regression (GBR) is a well-known supervised AI algorithm. It works efficiently with data with missing values and outliers, and this model can detect non-linear relationships between features quite well [49]. GBR is one of the most important ensemble methods that depend on obtaining an excellent prediction by collecting the prediction results of several poor methods together. It is an algorithm

that reduces the bias error of the model. In regression-related problems, this algorithm uses the mean squared error (MSE) as a cost function [50]. Gradient boosting is a very accurate prediction technique [51].

This algorithm is based on ensemble techniques that rely on grouping several decision trees [52]. The design of decision trees is similar to the design of a natural tree. Where it begins with the root and then the branches until it reaches the leaves, as the terminal leaf is considered the final result or goal. A disadvantage of decision trees is the overfitting of test data if the hierarchy is too deep [53]. To avoid such a problem, the GBR algorithm works to merge several decision trees with a technique somewhat similar to the technique used in the Random Forest. Random Forests generate multiple decision trees by randomly segmenting the data used. It avoids overfitting by obtaining the prediction of all individual decision trees and averaging the results.

On the other hand, GBR works by recursively adding decision trees so that the next decision tree corrects the error of the previous decision tree [54]. However, the results of this method are more sensitive to parameter settings during training than Random Forest. To get results better than Random Forest, correct parameter settings are used.

2.3.11 Elastic Net Regression

Linear regression assumes a linear relationship between the input variables and the target variable. To make linear regression more efficient and less affected by outliers, penalties were added to the loss function to encourage simpler models. After making these developments, two types

of linear regression appeared, which are penalized linear regression and regularized linear regression.

An Elastic Net is a well-known type of regular linear regression that relies on two important penalties, namely the penalty functions L1 and L2 [55]. That is, this algorithm is a combination of two techniques: Ridge and Lasso. As mentioned earlier, Lasso uses the L1 penalty and Ridge uses the L2 penalty. The main goal of Elastic Net is to reduce the following loss function [56]:

$$\frac{\sum_{i-1}^{n}(y_i - x_i'\hat{\beta})^2}{2n} + \lambda\left(\frac{1-\alpha}{2}\sum_{j-1}^{m}\hat{\beta}_j^2 + \alpha\sum_{j-1}^{m}|\hat{\beta}_j|\right) \quad (2.4)$$

As the value of α approaches 0, then Elastic is closer in its work to a Ridge, and vice versa, if the value of α approaches 1, then Elastic is closer in its work to a LASSO.

One of the advantages of Elastic Net is the improvement of lasso's limitations since lasso uses only a few samples for high-dimensional data [57]. As for the elastic network, it includes the "n" number of variables until saturation. The lasso tends to choose one variable if the variables are highly correlated groups and ignores the rest completely [58]. To avoid the limitations in the LASSO algorithm, the Elastic Net uses a quadratic expression ($|\beta|^2$) in the penalty. The quadratic expression elevates the loss function to be convex [59]. Thus, the elastic network depends on the best of the two algorithms, LSSO and Ridge regressions.

2.3.12 Bayesian Ridge

Bayesian is a technique used to define and estimate statistical models [60]. The benefit of Bayesian regression appears when the data is not well distributed or when the data is not sufficient at all. This algorithm is based on the probability distribution to get the prediction [61]. The objective 'y' is obtained from a normal distribution (where the mean and variance are normalized). Bayesian Regression aims to find the "posterior" distribution of the model's parameters. Posterior is the probability of an event occurring as a result of another event that has already occurred [62]. This is equivalent to Bayes' theorem which states [63]:

$$P(A \mid B) = \frac{P(B|A)\,P(A)}{P(B)} \tag{2.5}$$

Here, the value of P(A) represents the probability that event A will occur, and P(A | B) is the probability that event A will occur because event B has already occurred.

Based on the above formula, noticed that there is a subsequent distribution of the model parameters. This distribution is proportional to the probability of multiplying the data by the previous probability of the parameters. It is worth noting that increasing the number of data points leads to a significant increase in the probability value compared to the previous value. When this algorithm works to cover more data points, the built model becomes less erroneous. So, Bayesian Ridge needs a large amount of training data to make the model more accurate. For a fully probabilistic model with Bayesian regression, a Gaussian distribution

around X_w is used to obtain the output "y" as shown in the equation below [64]:

$$p(y|X,w,\alpha) = \mathcal{N}(y|Xw,\alpha) \tag{2.6}$$

Where alpha is treated as a random variable estimated from the data. Bayesian Ridge estimation is a probabilistic model of the regression problem. As indicated in the following equation [64]:

$$p(w|\lambda) = \mathcal{N}(w|0,\lambda^{-1}\mathbf{I}_p) \tag{2.7}$$

Where α and λ are chosen to be the gamma distributions. The model used for this equation is called Bayesian Ridge Regression, and it is efficient for low-dimensional data. It is similar to the classic Ridge model.

2.3.13 Kernel Ridge

The Kernel ridge regression (KRR) algorithm works by combining both the functions of classical ridge regression and classification (linear least squares with L2-norm regularization) on the one hand and the kernel trick on the other [65]. Where this algorithm learns a linear function in the resulting space from the kernel and the respective data.

The shape of the model learned by the KRR is similar to the shape of the model produced by the Support Vector Regression (SVR) algorithm [66]. However, SVR differs from KRR by the loss functions used, since KRR works with squared error loss while SVR uses loss Ɛ-insensitive, and both operate with L2 regularization. Kernel Ridge fit is performed in

a closed form, unlike SVR, in which KRR is faster on medium-sized datasets [67]. In contrast, the model learned from the KRR is slower than SVR because it is not sparse [68].

The kernel trick works well with linear data provided the number of features is large and the number of inputs is small [69]. What distinguishes the kernel ridge regression algorithm is the existence of formulas that enable the computation of the mean squared error of the leave using the results obtained from a single individual training over the entire training set, i.e. without performing the leave-one-out.

Thus, efficient hyper-parameters (ridge and kernel parameters) can be obtained. In addition, if a single value decomposition operation is performed, solutions matching many values can be calculated using a single training edge [70]. Applying this mechanism makes ridge improvement highly effective.

2.4 Related Work

Many researchers have worked hard over the years to study and develop food sales forecasting models. All of these researches and studies were to serve science and humanity, facilitate his work and reduce losses resulting from spending and wasting products. The researchers turned to AI algorithms, including DL algorithms, to achieve the best possible results. They used classification and Regression and developed existing algorithms using modern methods. Also, companies have become more flexible by providing the necessary datasets for studies to serve researchers and companies themselves. The most important related works can be summarized in the following paragraphs:

- To keep agricultural products fresh and ensure that the products are not spoiled, X. Wang, et al. suggested this model. The suggested model is based on integrating the most important weather factors affecting the sales of perishable agricultural products. Whereby, using the previous sales data for these products, three algorithms Random Forest, and Support Vector Machine (SVM), perform the required prediction. The results showed (according to the opinion of the researchers themselves) that this proposed model may significantly improve the prediction results depending on the weather conditions affecting fresh produce. Where the results achieved from this study, using the RMSE "Root Mean Square Error", are 68.90%, 23.66%, and 59.52%. The algorithms gave prediction results using MAE "Mean Absolute Error" as follows: 66.2%, 34.99%, and 61.13%, respectively [71].

- V. Prabhakar, et al. proposed in this study a model used to predict the future sales of a competing and leading grocery store. Where the store provided researchers with a dataset of food sales because they think it deserves study due to its importance and the benefit of the store itself. The store is guaranteed an opportunity to study and manage products that are subject to waste and damage. This study relies on ML algorithms to make an efficient prediction of food sales for this grocery store. The researchers used the R language to generate the statistics needed to predict the use of specific ML algorithms. Where they used a

library called Especially a caret to perform the prediction and analysis process [72].

- Forecasts have always been one of the important things in supply chains, now they have become a necessity, as a result of increasing consumer demand, limited capital, and the importance of dealing with limited time. Any store selling now wants to anticipate the products that consumers need to avoid seasonal shortages, so companies are working to develop forecasting daily. With the aforementioned, Y. F. Akande, et al. suggested using the "*extreme gradient boosting*" (XGBoost) algorithm to predict future sales. Sales data provided by 45 Walmart stores were used to develop the model. Where the results showed (according to the researcher) that the XGBoost ML algorithm proved effective in terms of predicting future sales [73].

- Predicting future sales of groceries increases merchant revenues by avoiding spoilage of surplus products as well as benefits the customer. Likewise, it is important to consider the issue of features affecting sales. For the above, Y. Liu suggested designing a model to make the necessary forecast for sales of large stores. This model is based on the use of two-time series cores and a light gradient boosting machine learning algorithm (LGBM) mainly. Where data is processed, analyzed, trained, tested, and then used. The resulting accuracy of this model was

measured using the mean squared error, which gave an accuracy of 0.35 [74].

- To facilitate the work of the staff in restaurants to meet the needs of customers, these restaurants need a good sales forecasting program. Using real data from a medium-sized restaurant, A. Schmidt, et al. proposed using several ML algorithms to make the appropriate prediction. Where recurrent neural network (RNN) technology was used, as well as three different datasets were used to train and test the models. The results obtained from the samples were compared for one day as well as for one week. The results for the one-day linear models using "*symmetric mean absolute percent error*" sMAPE were only 19.6%. When using RNN with "*long short-term memory*" LSTM and "*Temporal Fusion Transformer*" TFT algorithms, the results are good with errors of less than 20%. When performing the one-week forecasting process with the models without using RNN the result was bad, the results were approximately 20% error. With the use of RNN, the results using the sMAPE scale gave approximately 19.5% of the best result [75].

- T. S. Yange, et al. suggested building a model using the (SVM) algorithm to forecast the sales of agricultural products so that managers can determine the quantities needed for the sale of products that may be subject to spoilage. This system used both SVMs and "Fuzzy Theory". The Radial Basis Function (RBF)

neural network was taken as a standard for evaluating the result obtained from this model. To train the system, the data provided by "Makurdi University Farm" was used. The system was trained using one part of the data and was tested using another part. The SVM algorithm gave a result of 96.75% and the prediction result of the RBF was 90.55%. From this, the SVM algorithm gave the best results [76].

- T. Tanizaki, et al. suggested using POS data as internal data and adopting weather data, events data, and other external data to forecast sales. Where the algorithms of "Bayesian linear regression, decision tree regression, decision forest regression, and stepwise method" were used to perform the prediction process. The results showed closeness to the accuracy obtained from "Bayesian, Decision, and Stepwise", and the results obtained from Boosted were somewhat low. Overall, the expected rate of prediction exceeded 85% [77].

- In this study, T. Weng, et al. designed a developed model based on "LGBM and LSTM (Long-Short Term Memory)" for sales forecasting. Three databases were used to verify the accuracy and efficiency of the model. According to the researchers in this study, the model works efficiently to forecast supply chain sales. According to the researchers in this study, their model provides a high possibility of predicting long-term sales of the supply chain, which is beneficial for companies. The model is based not only on the efficiency of

LSTM but also on the possibility of LGBM, which is efficient in an industrial production environment [78].

- I. Vallés-Pérez, et al. developed a prediction system based on three day/store/item alternatives based on deep learning algorithms and their application on “Corporación Favorita” data. According to the results of this study, it provides good prediction accuracy by adopting a simple sequence of geometry sequences to pre-process the data. Also, they used a "training trick" this is to create a model less dependent on time and thus ensure good generalization over time. The model gave results estimated at 0.54 using RMSLE “Root Mean Squared Logarithmic Error” [79].

- To predict the future sales of each product accurately, X. Bi, et al. proposed a new model. Relying on “Tensor Methodologies for Context-Aware Recommendation Systems” this model was proposed under the name “Advanced Temporary Latent Factor Approach to Sales Forecasting, or ATLAS.” According to the researchers, this model achieves efficient prediction results for each product separately, by building a model of One tensioner worker for stores and products. The proposed model mixes two components of a "tensor framework" (depending on product and stores information), and the second component is the integration of demand mechanics, tensor extrapolation based on the latest (seasonal) statistical data for certain periods as well as using "seasonal autoregressive

integrated moving-average models" and RNN models. The main focus of the ATLAS model is the use of eight datasets. The model analyzes data from more than 1,500 grocery stores [80].

- Prediction models relied on ML algorithms and deep learning algorithms. However, to achieve higher accuracy and efficiency, other matters must be taken into account, such as epidemics, natural disasters, etc., which is reflected in the purchase of products For the above, N. Kumar, et al. suggested designing a new model that adopts a “multi-modal network” to forecast product sales based on the principle of combining current events with previous data. Moreover, the form also obscures the data collection published by Google Trends. The accuracy obtained from the preliminary results using "symmetric mean absolute percentage error SMAPE" showed approximately 7.37% compared to the results obtained from the current sales forecasting techniques [81].

- High promotions produce high sales that are difficult to deal with and manage, especially in how these sales are scheduled and their quantities. To mitigate such problems good sales forecasting models are used. On this basis, H. Dai, et al. suggested designing a model based on the random forest (RF) algorithm based on clustering to forecast sales on peak days. A dataset provided by a major grocery store in China was used. The algorithm gave 13.43% better prediction results than the non-clustered RF algorithm, and 22.85% better than the non-

clustered autoregressive integrated moving average (ARIMA) model [82].

- This study was proposed by C.-H. Wang and aims to (1) rely on economic indicators and their associated timelines as two basic elements for forecasting, (2) use both ML algorithms and deep learning algorithms to make the forecasting process, and (3) the dynamic interactions between competing companies from on the one hand, and the sales sector on the other. Preliminary results show that the sales revenues of the retail sectors are affected mainly by the retail employment census, real wages, as well as the consumer price index. As for seasonal factors, they affect hypermarkets only. The results showed that learning deep learning algorithms give the best results [83].

- Given the transition to the era of big data, R. Odegua proposed in this study a prediction model using the data of a store called "Chukwudi Supermarkets", with the application of three machine learning algorithms "K-Nearest Neighbor, Gradient Boosting, and Random Forest". The obtained results showed that the Random Forest algorithm gives better accuracy than the Gradient Boosted and K-Nearest Neighbor algorithms. To improve the accuracy of prediction, the model focused on three main variables: the type of store, the price of the product, and the year in which the store was opened [84].

- In this study, A. P. Wellens, et al. proposed a model based on the decision tree. A dataset containing 4,523 products from a leading retail store in Belgium was used and some external events such as national holidays and promotions were taken into account. The results gave an improvement in prediction accuracy over commonly used statistical methods by 9.34% and up to 20.52%. The proposed model relied on external variables, which improved the results obtained [85].

- There is no way to accurately predict sales during promotions. Because of this problem, J. Wolters and A. Huchzermeier proposed a model that deals with products during seasonal offers. The model is based on two stages. In the first stage, the seasonal sales cycle is predicted by using the harmonic regression model, and this is done by not using promotional sales data during the holiday period in the second stage, the results of the first stage are combined with a function that uses holidays and promotions data. Then the final model is formed using the ridge regression algorithm. The data provided by the grocery store is used for short periods as well as for long periods. This model gave good accuracy in forecasting sales [86].

- To forecast customer requirements in "Supply Chain Management SCM", N. Vairagade, et al. tried to use machine learning algorithms to make an appropriate prediction. A "representative set of ML-based prediction techniques" has been applied to the used dataset. Then "R^2, Mean Squared Error

(MSE), and MAE" scores were used to evaluate the accuracy of the algorithms used. Based on the results, the Random Forest algorithm gives a better prediction result compared to the artificial neural network that was applied to the same data [87].

- In this study, V. Adithya Ganesan, et al. proposes a model that predicts food sales in advance for a retail store in India. The model is taught through the internet as well as features engineering is adopted, after that, a neural network algorithm is used to make the prediction process. The proposed model gives better accuracy than the results obtained from the traditional time series models, as well as it gives better accuracy than the corporate's current model by 7.7%. According to researchers working on this model, it leads to saving 170 units of food per day [88].

- In this study, L. Zhou, et al. gave an introduction to DL and how neural networks work, as well as how to train the model. In this study, a survey of data related to food sales was conducted, so the survey included calories in foodstuffs, types of foodstuffs, their sources of contamination, and other matters. The specific problems, the existing data sets, the applied neural networks, as well as the obtained accuracy, were studied with a comparison with other research. The result of the research indicates that deep learning is superior to traditional artificial intelligence algorithms [89].

- S. Nosratabadi, et al. suggested designing a model based on two main techniques in the prediction work of food products. The “adaptive network-based fuzzy inference system (ANFIS) and multilayer perceptron (MLP)” methods are used to enhance prediction models. Two variables are adopted, the first is agricultural production and the other is livestock production. Three factors were relied upon to evaluate animal production, which are slaughtered animals, live animals, and wealth yield. As for agricultural production, two variables were relied upon, which are the outcome and losses of agricultural production. The data for this model was from Iran. Where data on livestock and agricultural production from 1961 to 2017 were collected from the "FAOSTAT" dataset. 70% of the data is used to train the model and 30% is used to test it. The obtained accuracy revealed that the ANFIS model has a low error rate [90].

A brief explanation was provided above for the most important and recent studies regarding food sales prediction, where AI algorithms were used and food-related datasets were used.

Table 2.1: Summary of Related Works

No.	Authors	Year	Algorithms used	Techniques
1	X. Wang, D. Lin, W. Fan, and T. Wang	2018	Random Forest, Ridge Regression, and SVM.	ML with Ensemble Technique.
2	V. Prabhakar, D. Sayiner, U. Chakraborty, T.	2018	Gradient Boosting Method, Neural	ML with Ensemble Technique, and DL.

	Nguyen, and M. Lanham		Networks, Linear Regression, and SVM	
3	Y. F. Akande, J. Idowu, A. Misra, S. Misra, O. N. Akande, and R. Ahuja	2022	XGBoost	ML with Ensemble Technique.
4	Y. Liu	2022	LGBM	ML with Ensemble Technique.
5	A. Schmidt, M. W. U. Kabir, and M. T. Hoque	2022	RNN with LSTM and TFT	ML, and DL.
6	T. S. Yange, C. O. Egbunu, O. Onyekwere, and K. A. Foga	2020	SVM, RBF	ML, and DL.
7	T. Tanizaki, T. Hoshino, T. Shimmura, and T. Takenaka	2019	"Bayesian linear regression, decision tree regression, decision forest regression, and stepwise method"	ML with Ensemble Technique, and DL.
8	T. Weng, W. Liu, and J. Xiao	2020	LGBM and LSTM"	ML with Ensemble Technique.
9	I. Vallés-Pérez, E. Soria-Olivas, M. Martínez-Sober, A. J. Serrano-López, J. Gómez-	2023	RNN	DL.

	Sanchís, and F. Mateo			
10	X. Bi, G. Adomavicius, W. Li, and A. Qu	2023	RNN	DL.
11	N. Kumar, K. Dheenadayalan, S. Reddy, and S. Kulkarni	2022	Multi-Modal network	DL.
12	H. Dai, H. Yu, Q. Xiao, and W. Zhou	2019	Random Forest	ML with Ensemble Technique.
13	C.-H. Wang	2022	Random Forest, Gradient Boosting, and ARIMA.	ML with Ensemble Technique, and DL.
14	R. Odegua	2020	"K-Nearest Neighbor, Gradient Boosting and Random Forest".	ML with Ensemble Technique.
15	A. P. Wellens, R. N. Boute, and M. Udenio	2022	Decision Tree	ML.
16	A. P. Wellens, R. N. Boute, and M. Udenio	2021	Ridge regression, and Harmonic regression	ML.
17	N. Vairagade, D. Logofatu, F. Leon, and F. Muharemi	2019	Artificial neural network and Random Forest	ML with Ensemble Technique, and DL.

18	. Adithya Ganesan, S. Divi, N. B. Moudhgalya, U. Sriharsha, and V. Vijayaraghavan	2019	Neural Network	DL.
19	L. Zhou, C. Zhang, F. Liu, Z. Qiu, and Y. He	2019	Neural Network	DL.
20	S. Nosratabadi, S. Ardabili, Z. Lakner, C. Mako, and A. Mosavi	2021	ANFIS and MLP	ML, and DL.

If noticed these studies, most of them adopted one dataset and a few AI algorithms, and include in Table 2.1 above the preparation of datasets for each study and the algorithms used, including ML algorithms, DL algorithms, as well as models that rely on ensemble techniques (techniques that are based on merging several models instead of using a single model to obtain greater prediction accuracy) so that the reader can notice what was used in these studies and compare it with what will discuss in this study.

CHAPTER THREE

PROPOSED METHOD

3.1 Overview

This study proposes a food sales prediction strategy. This strategy works to predict the quantities of products that companies need in the coming period under certain conditions and according to specific prices to reduce or prevent spoilage of foodstuffs as a result of their accumulation without selling them due to lack of demand. Several successive steps make up this model, which is illustrated in Figure 3.1.

The first step is to feed the model with three different datasets with varying degrees of correlation between their features, which will be discussed in detail in this chapter. The second step is to present the necessary charts and figures that give an initial view of the data used simply and clearly. This step is followed by the step of pre-processing to obtain suitable datasets ready for processing. The fourth step is to measure the degree of correlation between the features of the datasets by using the Heatmap tool from the Seaborn library available in the Python language. In the fifth stage, the three datasets are entered into thirteen algorithms used in this model. The sixth stage of this system is the use of several important metrics to measure the accuracy and efficiency of the algorithms used for each dataset separately and represent them graphically using simplified forms.

The seventh and final step is to test the proposed model through the use of a realistic dataset for sales of Iraqi Dates for the period (2002-2020), where the correlation of the features of this dataset is measured and according to the degree of correlation, the three appropriate algorithms are used based on the proposed results obtained from the model to give a prediction of the quantities of Dates that Iraq will produce in the five years following 2020, to sell them in the local and foreign markets.

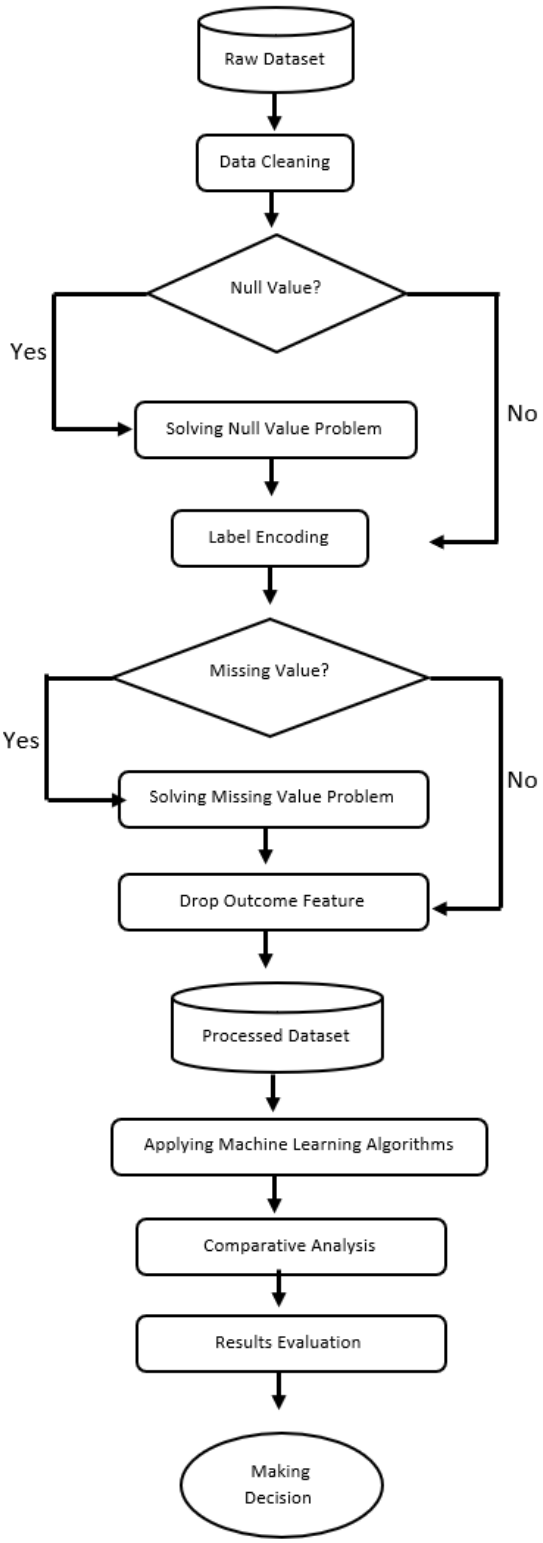


Figure 3.1: Proposed Framework.

3.2 Datasets Used

This study used four datasets, three of which are entered into the model to be used to clarify the impact of data features correlation on prediction results, as well as to know the best algorithm in terms of accuracy and efficiency in forecasting sales of food products that are applied to each dataset separately. As for the fourth dataset, the model is used to generate a future prediction of the quantities of foodstuffs that can be sold according to specific variables.

The first dataset is sourced from "*publicly available Alibaba's Tianchi platform data*" and it consists of 1000 rows and 15 columns as shown in Table 3.1. This data set contains food sales of different categories on the Alibaba platform in several cities in the year 2019.

Table 3.1: A Part of the First Dataset

	Invoice ID	Branch	City	Customer	Gender	Product line	Unit price	Quantity	Tax 5%	Date	Time	Cost	gross income	Rating	Total
2	765-26-69	A	Yangon	Normal	Male	Candy	72.61	6	21.783	1/1/2019	10:39	435.66	21.783	6.9	457.443
3	530-90-98	A	Yangon	Member	Male	Drinks	47.59	8	19.036	1/1/2019	14:47	380.72	19.036	5.7	399.756
4	891-01-70	B	Mandalay	Normal	Female	Fruites	74.71	6	22.413	1/1/2019	19:07	448.26	22.413	6.7	470.673
5	493-65-62	C	Naypyitaw	Member	Female	Candy	36.98	10	18.49	1/1/2019	19:48	369.8	18.49	7	388.29
6	556-97-71	C	Naypyitaw	Normal	Female	Fruites	63.22	2	6.322	1/1/2019	15:51	126.44	6.322	8.5	132.762
7	133-14-72	C	Naypyitaw	Normal	Male	Dairy products	62.87	2	6.287	1/1/2019	11:43	125.74	6.287	5	132.027
8	651-88-73	A	Yangon	Normal	Female	Biscuit	65.74	9	29.583	1/1/2019	13:55	591.66	29.583	7.7	621.243
9	182-52-70	A	Yangon	Member	Female	Candy	27.04	4	5.408	1/1/2019	20:26	108.16	5.408	6.9	113.568
10	416-17-99	A	Yangon	Member	Female	Fruites	74.22	10	37.11	1/1/2019	14:42	742.2	37.11	4.3	779.31

The second dataset was taken from *Kaggle*, and it is related to food sales for ten stores in different places, and it consists of 8523 rows and 12 columns as shown in Table 3.2.

Table 3.2: A Part of the Second Dataset

	Item_Id	Item_Weight	Fat_Content	Item_Visibility	Item_Type	Retail_Price	Store_Id	Store_Establishment_Year	Store_Size	Store_Location	Store_Type	Total_Sales
2	FDA15	9.3	Low Fat	0.016047301	Dairy	249.8092	OUT049	1999	Medium	Tier 1	Supermarket Type1	3735.138
3	DRC01	5.92	Regular	0.019278216	Soft Drinks	48.2692	OUT018	2009	Medium	Tier 3	Supermarket Type2	443.4228
4	FDN15	17.5	Low Fat	0.016760075	Meat	141.618	OUT049	1999	Medium	Tier 1	Supermarket Type1	2097.27
5	FDX07	19.2	Regular	0	Fruits and Ve	182.095	OUT010	1998		Tier 3	Grocery Store	732.38
6	NCD19	8.93	Low Fat	0	Household	53.8614	OUT013	1987	High	Tier 3	Supermarket Type1	994.7052
7	FDP36	10.395	Regular	0	Baking Goods	51.4008	OUT018	2009	Medium	Tier 3	Supermarket Type2	556.6088
8	FDO10	13.65	Regular	0.012741089	Snack Foods	57.6588	OUT013	1987	High	Tier 3	Supermarket Type1	343.5528
9	FDP10		Low Fat	0.127469857	Snack Foods	107.7622	OUT027	1985	Medium	Tier 3	Supermarket Type3	4022.764
10	FDH17	16.2	Regular	0.016687114	Frozen Foods	96.9726	OUT045	2002		Tier 2	Supermarket Type1	1076.599

The third dataset is related to food sales for several main categories and sub-categories in different cities. This dataset is sourced from *Kaggle* and consists of 9995 rows and 10 columns attributes as shown in Table 3.3.

Table 3.3: A Part of the Third Dataset

	Order ID	Customer	Category	Sub Category	City	Order Date	Region	Sales	Discount	Profit
2	OD1	Harish	Oil & Masala	Masalas	Vellore	11/8/2017	North	1254	0.12	401.28
3	OD2	Sudha	Beverages	Health Drinks	Krishnagiri	11/8/2017	South	749	0.18	149.8
4	OD3	Hussain	Food Grains	Atta & Flour	Perambalur	6/12/2017	West	2360	0.21	165.2
5	OD4	Jackson	Fruits & Veggies	Fresh Vegetables	Dharmapuri	10/11/2016	South	896	0.25	89.6
6	OD5	Ridhesh	Food Grains	Organic Staples	Ooty	10/11/2016	South	2355	0.26	918.45
7	OD6	Adavan	Food Grains	Organic Staples	Dharmapuri	6/9/2015	West	2305	0.26	322.7
8	OD7	Jonas	Fruits & Veggies	Fresh Vegetables	Trichy	6/9/2015	West	826	0.33	346.92
9	OD8	Hafiz	Fruits & Veggies	Fresh Fruits	Ramanadhapuram	6/9/2015	West	1847	0.32	147.76
10	OD9	Hafiz	Bakery	Biscuits	Tirunelveli	6/9/2015	West	791	0.23	181.93

The fourth and final dataset was used to experiment with the proposed model to generate a prediction of the quantities of food products that could be sold for the next five years under specific variables. This dataset is for sales of Dates in the Republic of Iraq for the period from (2002-2020) and includes several features, including the quantity, prices, types of Dates, and the number of palm trees in this period and consists of 115 rows and 5 columns attributes. The source of this dataset is the Iraqi Ministry of Planning / Central Statistical Organization. Table 3.4 shows this dataset.

Table 3.4: A Part of the Fourth Dataset

year	type of dates	number of palm	production of palm (Ton)	price of dates (IQD)
2002	zahdi	9413000	69089	120
2002	khestawy	1047000	7033	200
2002	khadrawy	584000	1915	225
2002	sair	1864000	3629	270
2002	hellawy	721000	2642	200
2002	others	1229000	7125	300
2003	zahdi	7900000	55456	160
2003	khestawy	950000	4734	270
2003	khadrawy	431000	4449	260

3.3 Visualization

In this part of the chapter figures and diagrams will be presented to facilitate understanding of the contents of the datasets used. These diagrams will also show how some features affect each other, which paves the way for the next part of this chapter, which is concerned with studying the correlation of features of datasets. Where it will start with the charts of the first dataset, then the second, and finally the third.

Figure 3.2 shows the frequency of food categories in the first dataset. It is observed that there is a convergence in the distribution of numbers in this dataset.

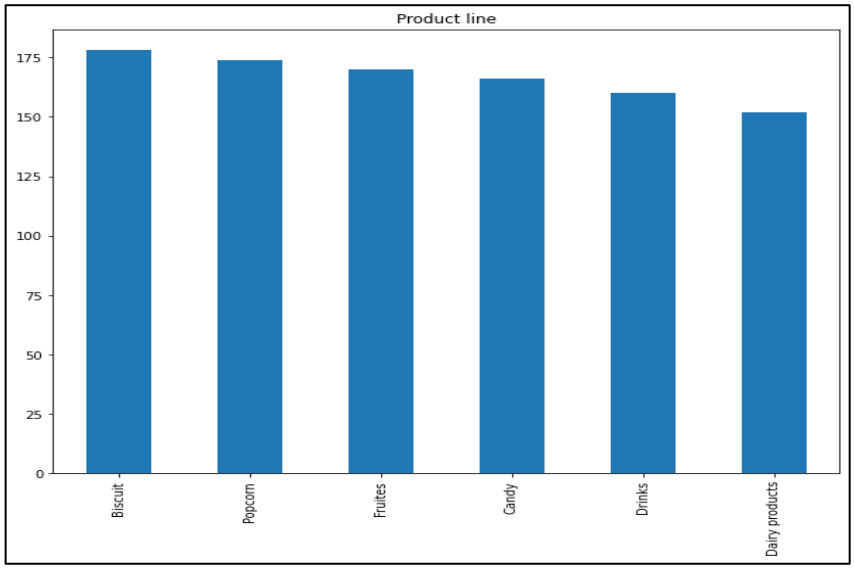


Figure 3.2:Frequency of Products Categories in the First dataset

Figure 3.3 shows the relationship between the quantities of products and their category, as the average quantity of products in the first dataset was taken.

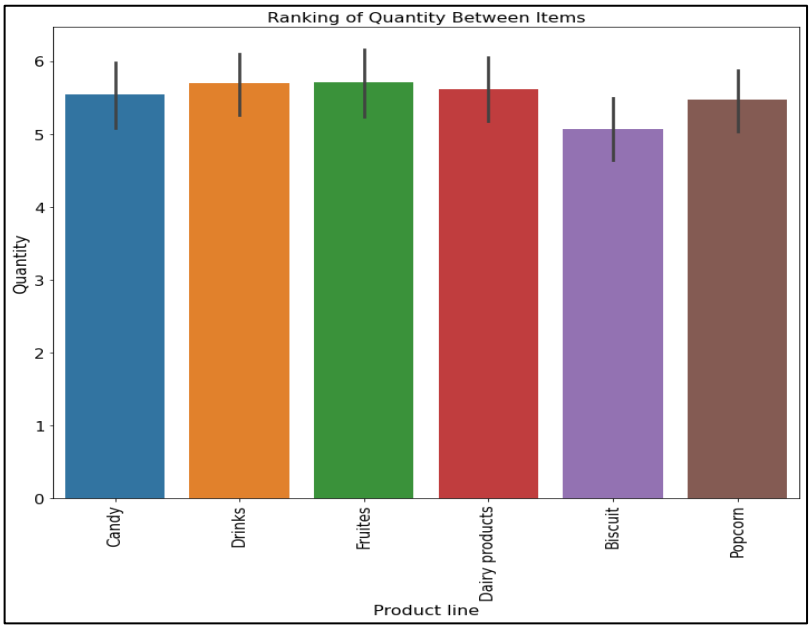


Figure 3.3: The Relationship between Product Category and Quantity in the First dataset

Figure 3.4 includes a 3D chart showing the relationship between product category, quantity, and total price of the first dataset.

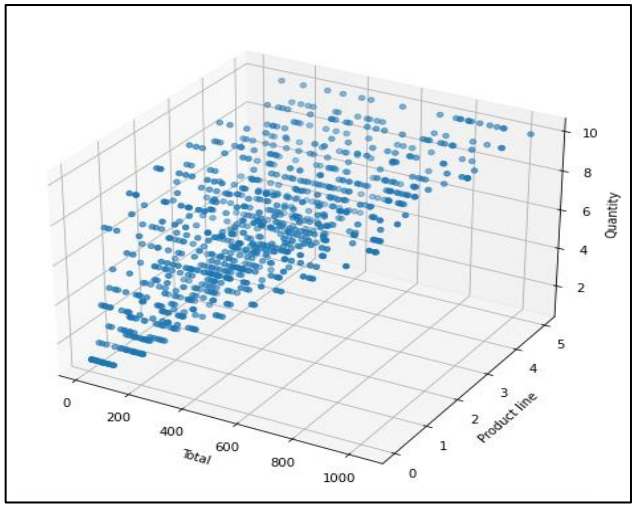


Figure 3.4: The Relationship between Product Category, Quantity, and Total price in the First dataset

Below are the charts for the second dataset, starting with Figure 3.5, which represents the frequency of food categories in this dataset.

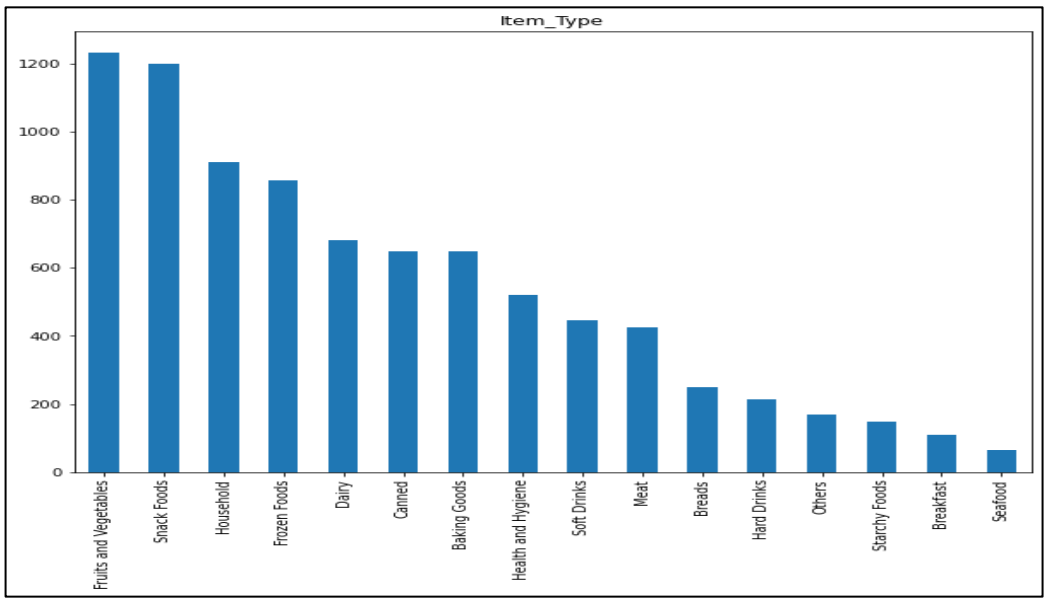


Figure 3.5: Frequency of Products Categories in the Second dataset

Figure 3.6 below shows the relationship between the type of product and its price in the second dataset. What is noticed is the convergence of prices between the categories.

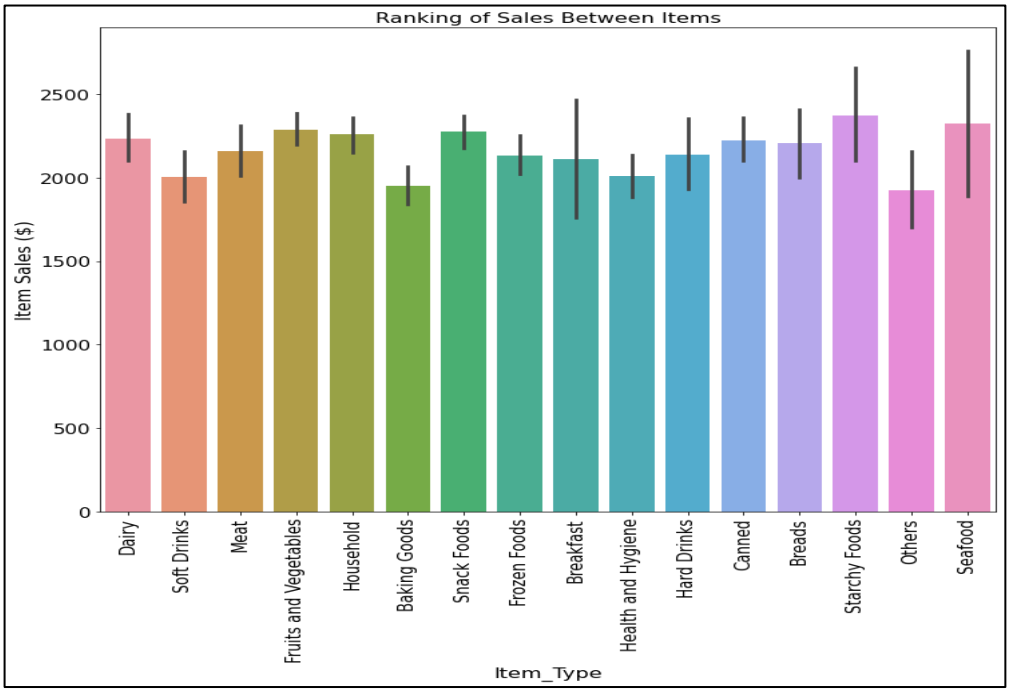


Figure 3.6: The Relationship between the Type of Product and the Price in the Second Dataset

Figure 3.7 below contains a three-dimensional chart showing the relationship between the type of product, the store offered in it, and the total price in the Second Dataset.

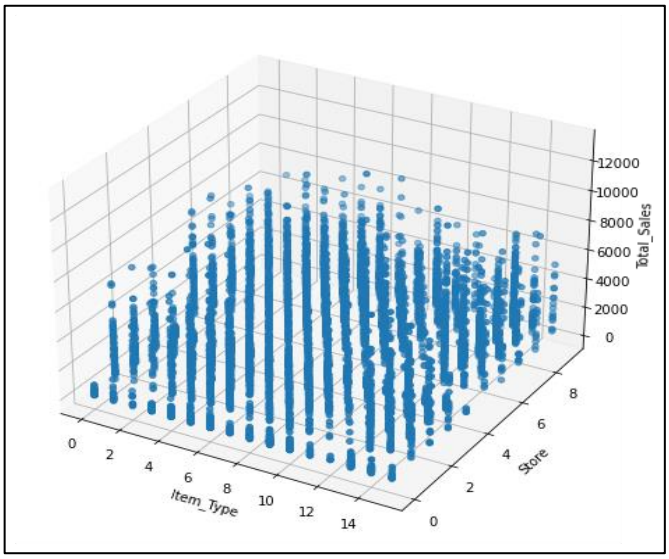


Figure 3.7:The Relationship between the Type of Product, the Store, and the Total Price in the Second Dataset.

Finally, below are the charts for the third dataset, which shed light on the most important relationships between variables to facilitate understanding of the data. Figure 3.8 shows the frequency of food products in this dataset.

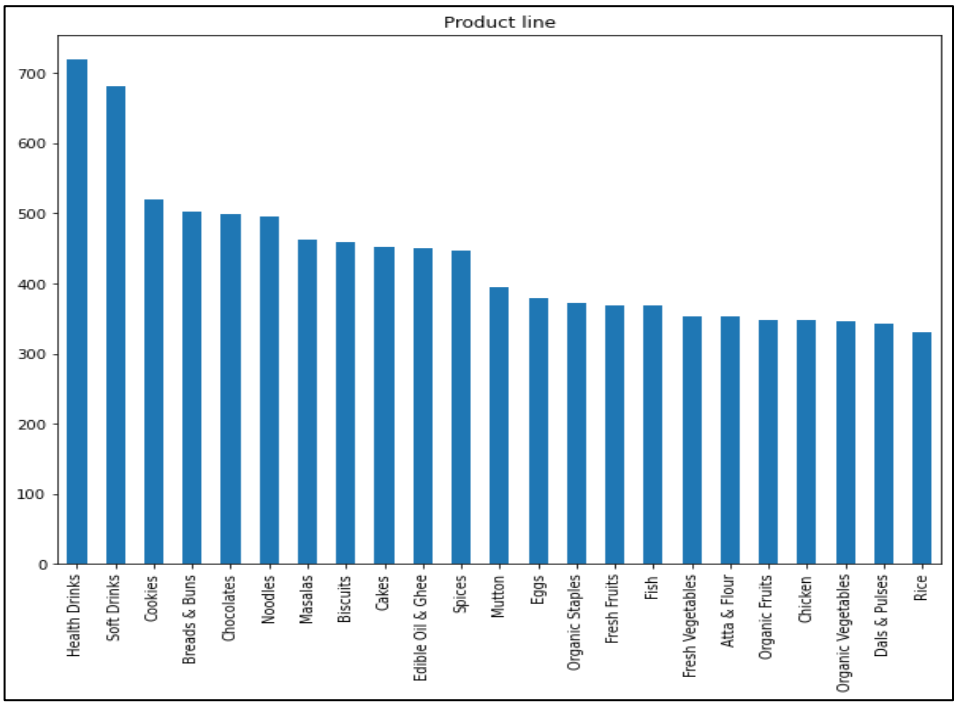


Figure 3.8: The Frequency of Food Products in the Third Dataset.

Figure 3.9 below shows the relationship between product type and quantity in the third dataset. Where the great diversity of food products and their convergence in quantities are noted.

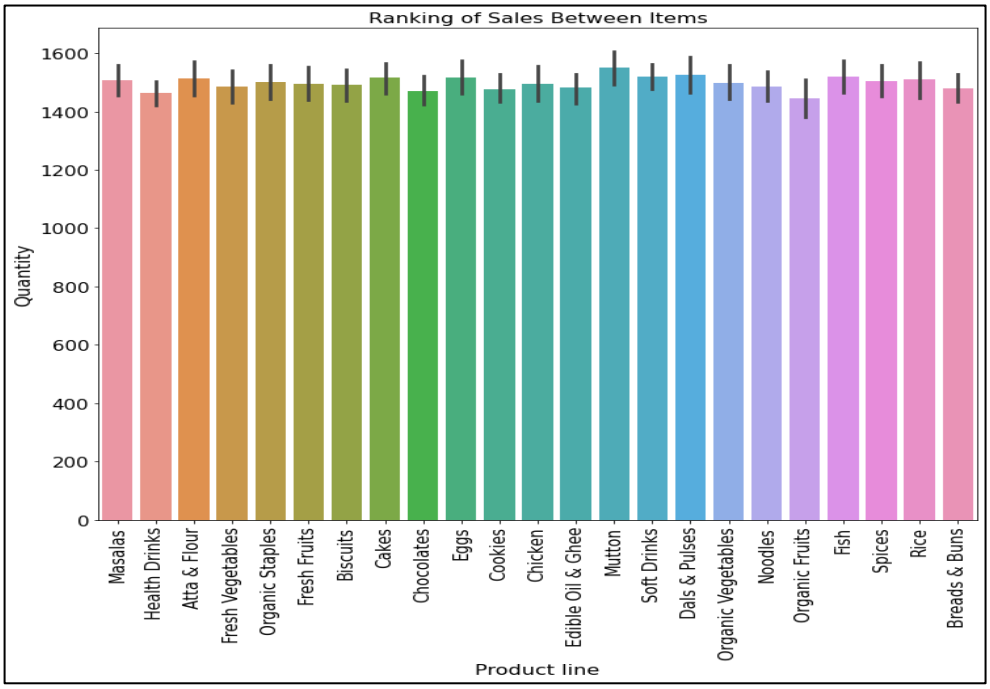


Figure 3.9: The Relationship between Product Type and Quantity in the Third Dataset

Finally, Figure 3.10 contains a three-dimensional chart that shows the relationship between the category of food product, its quantity, and the achieved price, which is the final profit.

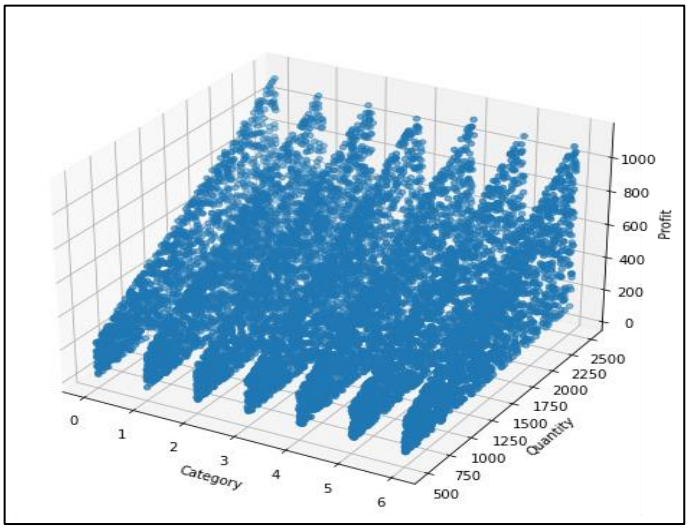


Figure 3.10: The Relationship between the Category of Product, its Quantity, and the Price in the Third Dataset.

3.4 Pre-processing

All necessary steps for pre-processing the data have been taken for each dataset to make them suitable for use. Below are the steps followed in preprocessing:

1. Processing human errors in data input. For example, in the 'Fat content' column of the second dataset, the term 'Low fat' was sometimes entered as 'LF,' so all entries were converted to 'Low fat'.

2. Handling missing data. For instance, in the 'Item weight' column, 1463 missing values were identified and replaced using the 'Imputer' algorithm with a mean strategy. The mean of the values in the relevant field was taken, and it replaced the missing values.

3. Processing textual inputs: Word categories were converted into numerical categories for handling as numeric values. Where the 'Label Encoder' algorithm was utilized for this purpose

In the end, good and organized datasets were obtained so that they could be dealt with. This study also sought not to cause large changes in the data and contents of these datasets to simulate the use of such datasets in other studies.

3.5 Features Correlation

To achieve a comprehensive view of the datasets used, three different datasets were taken into account. One of the most important

differences that distinguish one dataset from another is how the features are correlated. To find out how the features of the datasets are correlated, Correlation Matrix Heatmap was used. It is a tool available from the Seaborn library in the Python language. It is a two-dimensional matrix containing colored cells. The colors are graded according to the correlation strength. Whenever the color tends to become darker, this means a stronger correlation, and on the contrary, whenever it tends to become lighter, this means a weaker correlation. The cells of the matrix contain values between (-1 and 1), where whenever this value is directed to the positive, this indicates the strength of the correlation, and whenever it is directed to the negative, this indicates a weak correlation of the features.

Using the Correlation Matrix Heatmap tool, the correlation between the features of the three datasets used in this model was measured. Figure (3.11) below shows the correlation using this tool. What is noticed is that most of the cells tend to be dark in color, and there are many cells close to (1), which indicates that the first dataset is well correlated.

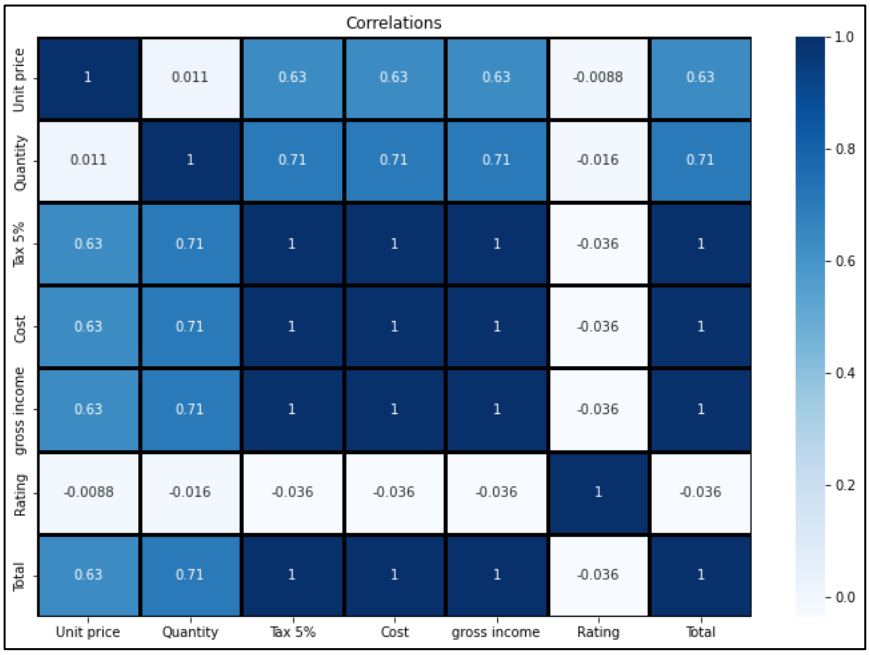


Figure 3.11: The Correlation between Features of the First Dataset.

In the same way, the correlation of features for the second dataset was measured. Figure (3.12) below shows that the matrix contains medium values. This indicates that the second dataset has a medium correlation between its features.

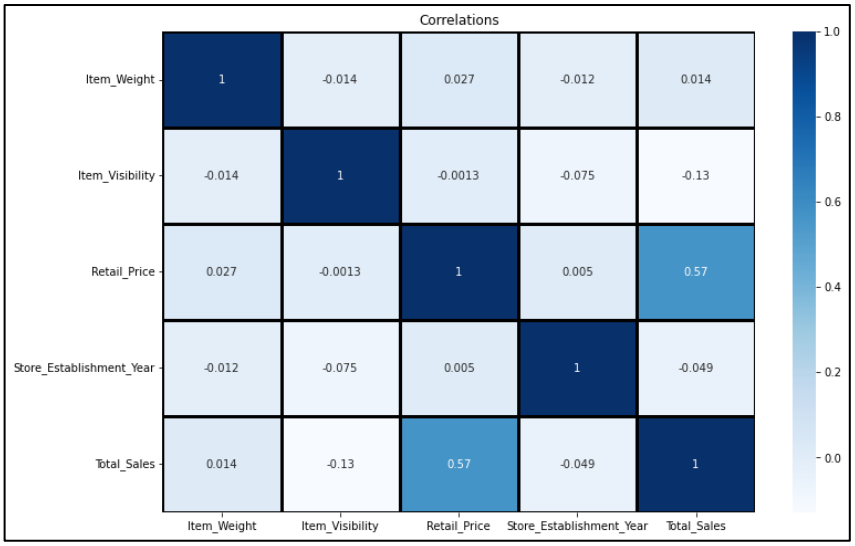


Figure 3.12: The Correlation between Features of the Second Dataset.

Finally, the correlation between the features of the third dataset was measured. Figure (3.13) below shows that. It is noted that most cell values tend to be negative, which indicates a weak correlation between the features of this dataset.

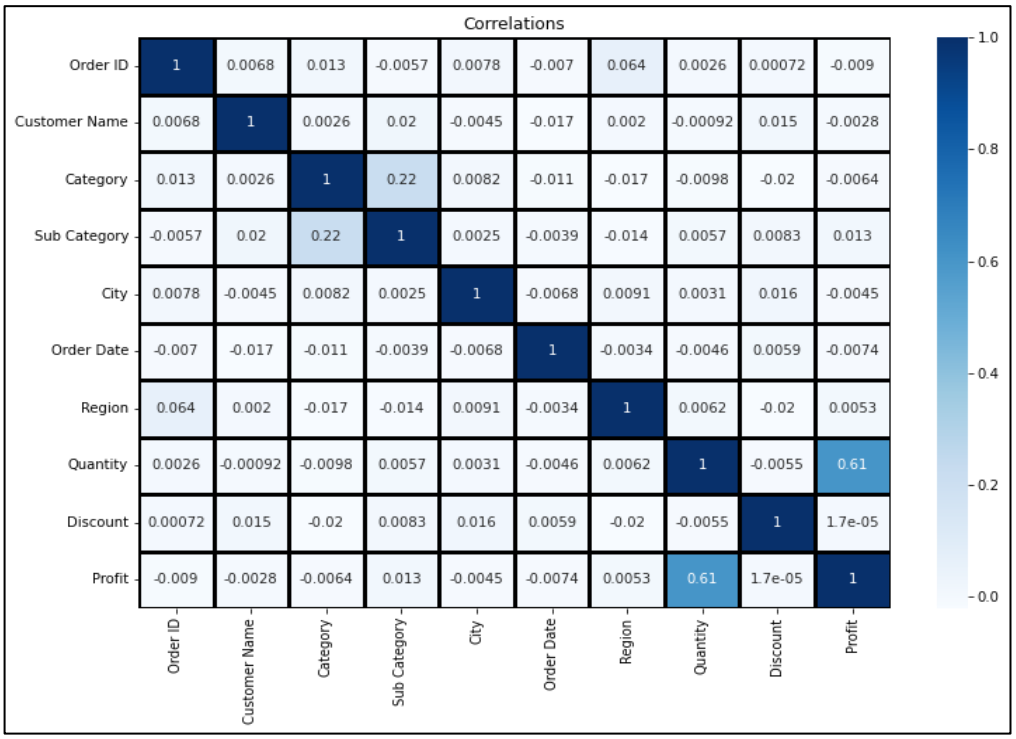


Figure 3.13: The Correlation between Features of the Third Dataset.

From what was mentioned above, the conclusion was reached that the first dataset is of strong correlation, the second is of medium correlation, and the third is of weak correlation. On this result, this study will be based. It will apply thirteen algorithms to each dataset separately to find out the best algorithm applied to each dataset.

3.6 Performance Measurements

One of the objectives of this study is to make an effective comparison between the most important ML algorithms used in forecasting food sales. To achieve this, several metrics were used to

measure the accuracy of these algorithms, including (R^2, MAE, and MSE):

- R-Squared (R^2) is a statistical equation that determines the percentage of variance in the dependent variable that can be predictable by the independent variable(s) [91]. This metric gives the proportion of the fit of the data to the regression model. Its value is between (−∞-1) and was multiplied by 100 in this study to know the percentage of accuracy obtained [92]. R^2 can be explained mathematically as follows [93]:

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(x_i - y_i)^2}{\sum_{i=1}^{m}\left(\left(\frac{1}{m}\sum_{i=1}^{m} y_i\right) - y_i\right)^2} \quad (3.1)$$

  Where x_i is the predicted i^{th} value, and the y_i element is the actual i^{th} value.

- The mean absolute error (MAE) is a metric used with regression models. Where it results in the average absolute difference between the expectation obtained from the model and the target value. MAE can be explained mathematically as follows [94]:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (3.2)$$

  Where n is the observation number, y_i is the observed value and $\hat{y}_i$ is the predicted value.

- Mean squared error (MSE) is a metric used with regression models. MSE represents how close the regression line is to the data points. where it is considered a risk function. The lower its value, the more accurate the model. MSE can be explained mathematically as follows [95]:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - Y_i')^2 \quad (3.3)$$

where n is the observation number, Y_i is the observed value and Y'_i is the predicted value.

3.7 Algorithms Used

A large number of AI models have been used. For comparison, many scales and diagrams were used. To ensure that the codes are not repeated, two algorithms were used below. The first algorithm produces the required metrics. The second algorithm produces the necessary diagrams for illustration. From the outputs of both algorithms, the accuracy obtained from the models can be compared.

First Algorithm

Input: s (model), X_train, X_test, y_train, y_test

Import necessary libraries
Import R2, MAE, MSE, and RMSE

# Make predictions on the test set
y_p = s.predict(X_test)

Display R2 score
Output: r2_score(y_test, y_p) * 100

```
# Evaluate and display training and test scores
Train_Score = s.score(X_train, y_train)
Test_Score = s.score(X_test, y_test)
Output: Train_Score, Test_Score

# Make predictions on training and test sets
trainPred = s.predict(X_train)
testPred = s.predict(X_test)

# Calculate and display MAE for training and test sets
MAE_Train = MAE(y_train, trainPred)
MAE_Test = MAE(y_test, testPred)
Output: MAE_Train, MAE_Test

# Calculate and display MSE for training and test sets
MSE_Train = MSE(y_train, trainPred)
MSE_Test = MSE(y_test, testPred)
Output: MSE_Train, MSE_Test

# Calculate and display RMSE for training and test sets
RMSE_Train = sqrt(MSE_Train)
RMSE_Test = sqrt(MSE_Test)
Output: RMSE_Train, RMSE_Test
```

End

Second Algorithm

```
Input: v (model), X_test, y_test

# Make predictions on the test set
y_p = v.predict(X_test)

# Create a DataFrame with 'Actual' and 'Predicted' columns
Data_Frame = Create_Data_Frame('Actual': y_test, 'Predicted': y_p)

# Display the DataFrame
Display(Data_Frame)

# Plot the data
Plot(Data_Frame, fig_size=(14,12))
```

```
    # Show the plot
    Show(plot)
End
```

3.8 Models Applied

The features used by the algorithm were defined as "X" and also to determine the feature that the algorithm should predict as "Y" and in this study, it represents the quantities of products the company needs. After that, the data was divided as a whole into train data, which is used to train the model and represents (80%), and into test data, in which the model is tested and represents (20%) of the total data.

This study is based mainly on the thirteen ML algorithms. To achieve a comprehensive comparison between most of the effective and well-known algorithms used in forecasting food sales in most studies, various ML algorithms were used, which can be classified into three different types: ML algorithms (SVR, KNN, Multiple Linear Regression, Decision Tree, Gaussian Processes Regressor, RANSAC, Ridge Regressor, Elastic Net, Bayesian Ridge, and Kernel Ridge), DL algorithm (Multilayer Perceptron), and models that rely on ensemble techniques (Random Forest Regressor, Bagging Regressor, Gradient Boosting). Below, it will be explained how these algorithms work by writing a pseudocode for each algorithm separately:

Multilayer Perceptron (MLP)

```
    # Input:
    # x_train: Training features matrix
    # y_train: Training target vector
```

```
# Import MLPRegressor
Import MLPRegressor

# Initialize MLPRegressor model
M = MLPRegressor()

# Fit the model on the training data
M.Fit(x_train, y_train)

# Output:
# - Model Metrics Scores
Metrics_Score(M)

# Output:
# - Visualization of the model
Visualization(M)
```

End

Support Vector Regressor (SVR)

```
# Input:
#  x_train: Training features matrix
# y_train: Training target vector

# Import SVR
Import SVR

# Initialize SVR model
S = SVR()

# Fit the model on the training data
S.Fit(x_train, y_train)

# Output:
# - Model Metrics Scores
Metrics_Score(S)
```

```
    # Output:
    # - Visualization of the model
    Visualization(S)
End
```

Multiple Linear Regression

```
    # Input:
    # x_train: Training features matrix
    # y_train: Training target vector

    # Import Multiple Linear Regression
    Import Multiple_Linear_Regression

    # Initialize Multiple Linear Regression model
    MLR = Multiple_Linear_Regression()

    # Fit the model on the training data
    MLR.Fit(x_train, y_train)

    # Output:
    # - Model Metrics Scores
    Metrics_Score(MLR)

    # Output:
    # - Visualization of the model
    Visualization(MLR)
End
```

Decision Tree

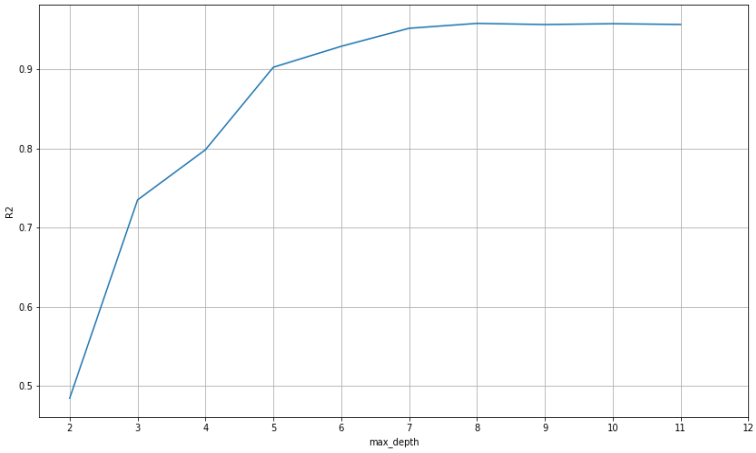
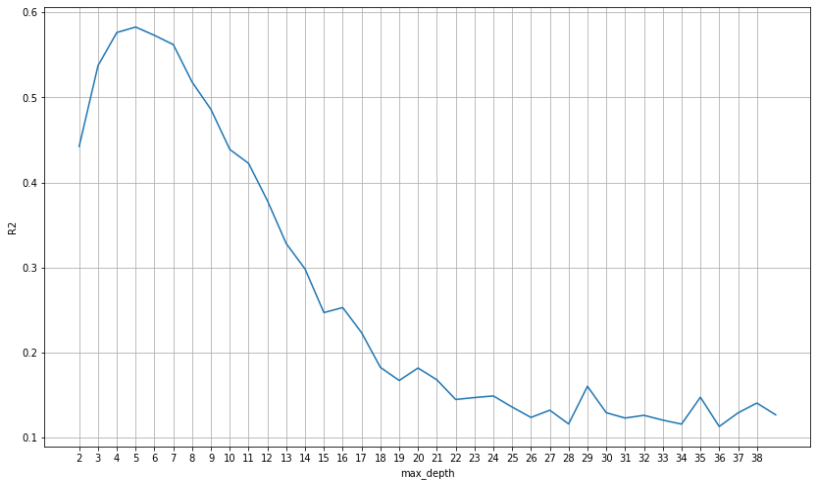
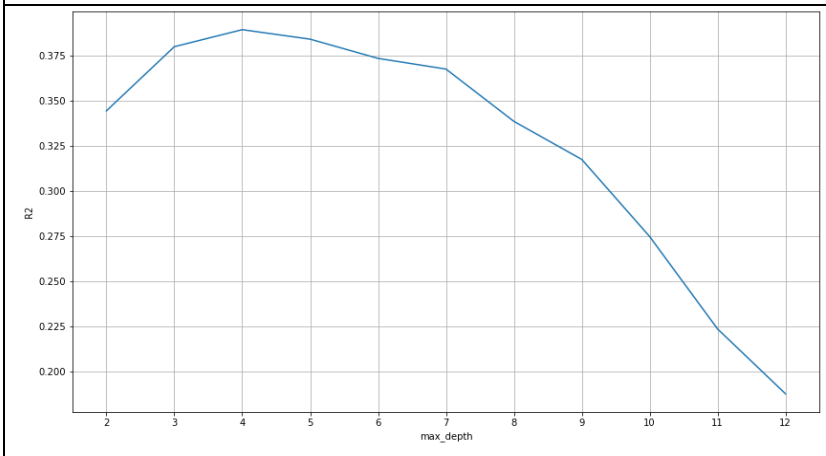
```
    # Input:
    # x_train: Training features matrix
    # y_train: Training target vector

    # Import Decision Tree
```

```
Import (Decision_Tree)

# Calculate best depth
maxDepthRange == list (range(2, 12))
for D in maxDepthRange
   { decTree == Decision_Tree (D == depth)
     decTree.fit (x1_train, y1_train)
     score = decTree.score (x1_test, y1_test)
     r2[]. Append (score) }
Plot (maxDepthRange, r2)

# Initialize Decision_Tree
decTree = Decision_Tree (max_depth)
# Fit the model on the training data
 decTree. Fit (x_train,y_train)

# Output:
# - Model Metrics Scores
 Metrics_Score (decTree)
# Output:
# - Visualization of the model
 visualization (decTree)
```

End

The decision tree algorithm is similar to real trees. Where it has several forks and a certain depth. To determine the best possible depth in which this algorithm can work, a special code has been written for this topic as shown in the pseudocode above. After that, the algorithm was applied to each dataset separately, as each dataset gave a different depth, and various results were obtained as well, as shown in the following tables:

Table 3.5: The Best Depth for the Decision Tree Algorithm with the Datasets

Dataset 1

Dataset 2

Dataset 3


Random Forest

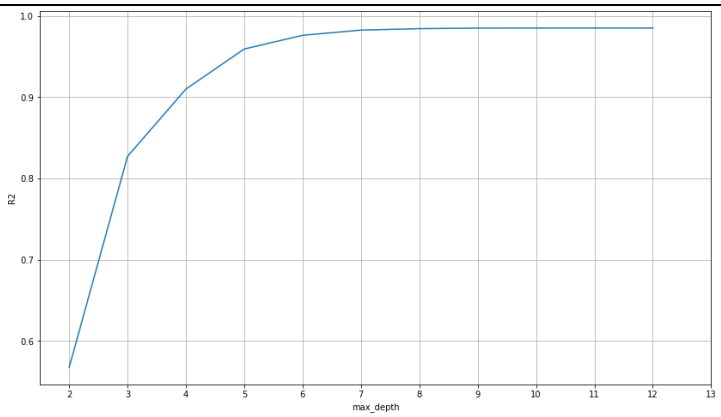
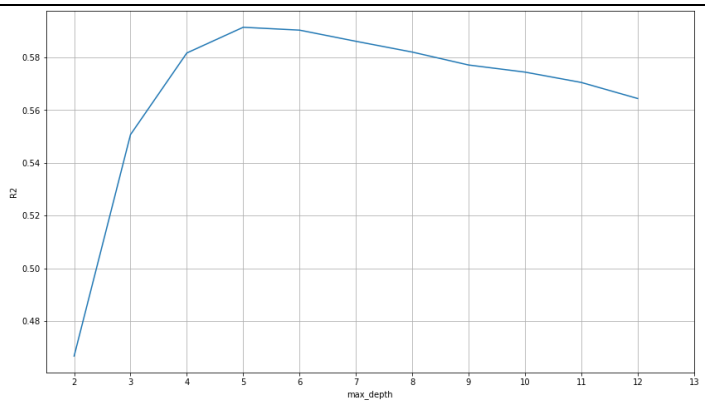
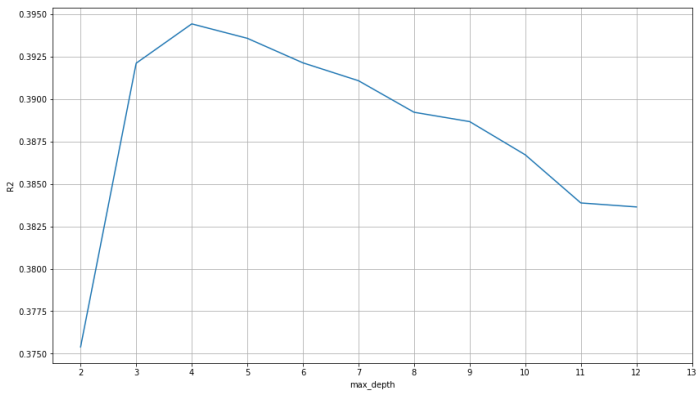
```
# Input:
# x_train: Training features matrix
# y_train: Training target vector

# Import Random Forest
Import (Random_Forest )
# Calculate best depth
maxDepthRange == list (range(2, 13))
for D in maxDepthRange
   { RF == Random_Forest (D == depth)
     RF.fit (x1_train, y1_train)
     score = RF.score (x1_test, y1_test)
     r2[]. Append (score) }
Plot (maxDepthRange, r2)

# Initialize Random_Forest
RF = Random_Forest (max_depth)

# Fit the model on the training data
 RF. Fit (x_train,y_train)

# Output:
# - Model Metrics Scores
 Metrics_Score (RF)
# Output:
# - Visualization of the model
 visualization (RF)
```

End

The random forest algorithm consists of several decision trees that combine to generate the best prediction. To determine the best depth to which the random forests reach, and at which they give the best prediction, a special code was used for this purpose, as shown in the pseudocode above. After that, the algorithm was applied to each dataset

separately, as each dataset gave a different depth, and various results were obtained as well, as shown in the following tables and figures:

Table 3.6: The Best Depth for the Random Forest Algorithm with the Datasets.

Dataset 1

Dataset 2

Dataset 3


K-Nearest Neighbors (KNN)

```
# Import the KNN library
from sklearn.neighbors import KNeighborsRegressor

# Create a KNN model
K = KNeighborsRegressor()

# Train the model using the training data
K.fit(x_train, y_train)

# Evaluate the model using metrics
metrics_score = Metrics_Score(K)

# Visualize the results using a visualization function
visualization(K)
```

End

Bagging Regressor

```
# Import the Bagging Regressor library
from sklearn.ensemble import BaggingRegressor

# Create a Bagging Regressor model
B = BaggingRegressor()

# Train the model using the training data
B.fit(x_train, y_train)

# Evaluate the model using metrics
metrics_score = Metrics_Score(B)

# Visualize the results using a visualization function
visualization(B)
```

End

Gaussian Processes Regressor

```
# Import the Gaussian Processes Regressor library
from sklearn.gaussian_process import GaussianProcessRegressor

# Create a Gaussian Processes Regressor model
G = GaussianProcessRegressor()

# Train the model using the training data
G.fit(x_train, y_train)

# Evaluate the model using metrics
metrics_score = Metrics_Score(G)

# Visualize the results using a visualization function
visualization(G)
```

End

RANSAC

```
# Import the RANSAC library
from sklearn.linear_model import RANSACRegressor

# Create a RANSAC model
R = RANSACRegressor()

# Train the model using the training data
R.fit(x_train, y_train)

# Evaluate the model using metrics
metrics_score = Metrics_Score(R)

# Visualize the results using a visualization function
visualization(R)
```

End

Gradient Boosting Regressor

```
# Import the Gradient Boosting library
from sklearn.ensemble import GradientBoostingRegressor

# Create a Gradient Boosting model
GB = GradientBoostingRegressor()

# Train the model using the training data
GB.fit(x_train, y_train)

# Evaluate the model using metrics
metrics_score = Metrics_Score(GB)

# Visualize the results using a visualization function
visualization(GB)
```

End

Elastic Net

```
# Import the Elastic Net library
from sklearn.linear_model import ElasticNet

# Create an Elastic Net model
E = ElasticNet()

# Train the model using the training data
E.fit(x_train, y_train)

# Evaluate the model using metrics
metrics_score = Metrics_Score(E)

# Visualize the results using a visualization function
visualization(E)
```

End

Bayesian Ridge

```
# Import the Bayesian Ridge library
from sklearn.linear_model import BayesianRidge

# Create a Bayesian Ridge model
BR = BayesianRidge()

# Train the model using the training data
BR.fit(x_train, y_train)

# Evaluate the model using metrics
metrics_score = Metrics_Score(BR)

# Visualize the results using a visualization function
visualization(BR)
```

End

Kernel Ridge

```
# Import the Kernel Ridge library
from sklearn.kernel_ridge import KernelRidge

# Create a Kernel Ridge model
KR = KernelRidge()

# Train the model using the training data
KR.fit(x_train, y_train)

# Evaluate the model using metrics
metrics_score = Metrics_Score(KR)

# Visualize the results using a visualization function
visualization(KR)
```

End

To summarize the topic and for all the parties to the conversation, an integrated summary of the obtained results will be developed, which allows an easy understanding of the differences between the datasets on the one hand, and the ML algorithms on the other hand. Also, one of the important things is to test the proposed model in this study. For this purpose, a real dataset on sales of Iraqi Dates was used. This topic will be discussed in detail in the fourth chapter.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Overview

This chapter will summarize all of the above in Chapter Three. Where it will deal with the results achieved from each algorithm used and for each dataset separately. Then the best algorithm that gave the highest prediction accuracy in the amount of food that can be sold will be selected. Hence, on this basis, the proposed model will be tested in this study through a new realistic dataset for sales of Iraqi Dates. The sequential steps of the model will be performed. The degree to which the data of that dataset is correlated is known. Then, based on the degree of correlation, the best algorithm is used to give the best prediction.

4.2 The Results of the Proposed Model

In the proposed model in this study, thirteen different algorithms were applied to three different datasets:

4.2.1 Multilayer Perceptron (MLP)

The above MLP algorithm gave the following results for each dataset separately:

Table 4.1: The Scores of MLP with the Datasets.

Dataset 1	Dataset 2	Dataset 3
DATA1 Accuracy of our model is 58.97 score for Training: 0.702715405556289 score for Testing: 0.5897348097522922 MAE for Training: 1.2384361841582783 MAE for Testing: 1.352727304303326 MSE for Training: 2.5796406175674202 MSE for Testing: 3.269813566274231 RMSE for Training: 1.6061259656600475 RMSE for Testing: 1.8082625822247804	DATA2 Accuracy of our model is 47.18 score for Training: 0.4683667696670468 score for Testing: 0.4718239937994978 MAE for Training: 929.8644023784533 MAE for Testing: 946.915958073672 MSE for Training: 1545993.2718168593 MSE for Testing: 1545896.936513265 RMSE for Training: 1243.3797777898992 RMSE for Testing: 1243.3410378947785	DATA3 Accuracy of our model is 35.58 score for Training: 0.3219613529595603 score for Testing: 0.3557555479490716 MAE for Training: 386.83042569292815 MAE for Testing: 375.19526486855204 MSE for Training: 225926.7104203931 MSE for Testing: 214547.73357631298 RMSE for Training: 475.3174838151792 RMSE for Testing: 463.1929766051219

Table (4.1) above shows the scores obtained from the MLP algorithm using various metrics and with the three datasets. Where it is noted that the first dataset with the highest feature correlation gave 58.97, followed by the second dataset with a medium correlation which gave 47.18, and then the third dataset with the weakest correlation which gave 35.58.

Table 4.2: The Actual and Predicted Values obtained by MLP.

Dataset 1

DATA1

	Actual	Predicted
993	8	7.663427
859	6	3.998882
298	6	7.363441
553	4	4.921832
672	7	6.256015
..	...	...
679	2	0.952432
722	3	3.494969
215	7	5.974847
653	3	2.445691
150	7	8.990973

[200 rows x 2 columns]

Dataset 2

DATA2

	Actual	Predicted
4931	1426.1436	2571.249978
4148	1201.7690	1923.881805
7423	1836.2764	1400.441734
4836	2410.8618	1606.616857
944	1549.9824	3009.971341
...	...	...
4644	3235.7880	2190.800507
6179	555.2772	2184.519895
1861	2885.5772	4147.612471
3598	218.3824	272.280336
1523	5478.2024	4965.270770

[1705 rows x 2 columns]

Dataset 3

DATA3

	Actual	Predicted
7933	610	1195.864115
8657	1712	1856.533956
9599	642	912.043587
799	2018	2020.540661
3813	1776	1752.619331
...	...	...
2890	829	1184.817859
4890	1539	1471.733328
2738	1817	1563.736685
5981	1658	1556.385744
6087	1626	2168.594116

[1999 rows x 2 columns]

Table (4.2) above shows the actual values and the values obtained from the MLP model. Where it is noted that the algorithm gives a better prediction with the first dataset compared to the second and third.

4.2.2 Support Vector Regressor (SVR)

The above SVR algorithm gave the following results for each dataset separately:

Table 4.3: The Scores of MLP with the Datasets.

Dataset 1	Dataset 2	Dataset 3
DATA1 Accuracy of our model is 82.29 score for Training: 0.8110079600532945 score for Testing: 0.8228738478082553 MAE for Training: 0.9211577771063892 MAE for Testing: 0.8377100861064218 MSE for Training: 1.6399488966312956 MSE for Testing: 1.4116954329682045 RMSE for Training: 1.2806048948177948 RMSE for Testing: 1.1881479002919648	DATA2 Accuracy of our model is 47.63 score for Training: 0.47325518576821346 score for Testing: 0.4762509654187005 MAE for Training: 906.1207653451144 MAE for Testing: 925.1295588361531 MSE for Training: 1531777.7225038265 MSE for Testing: 1532939.81278213 RMSE for Training: 1237.6500807998302 RMSE for Testing: 1238.1194662802657	DATA3 Accuracy of our model is 36.89 score for Training: 0.3360125103328011 score for Testing: 0.36888837017799614 MAE for Training: 375.21288877784247 MAE for Testing: 365.32863903257794 MSE for Training: 221244.77705746162 MSE for Testing: 210174.21163800743 RMSE for Training: 470.3666410976246 RMSE for Testing: 458.4476105707253

Table (4.3) above shows the scores obtained from the SVR algorithm using various metrics and with the three datasets. Where it is noted that the first dataset with the highest feature correlation gave 82.29, followed by the second dataset with the medium correlation which gave 47.63, and then the third dataset with the weakest correlation which gave 36.89.

Table 4.4: The Actual and Predicted Values obtained by SVR.

Dataset 1	Dataset 2	Dataset 3
DATA1 Actual Predicted 993 8 7.944878 859 6 5.757806 298 6 6.154858 553 4 4.528363 672 7 6.947956 .. .. 679 2 3.933126 722 3 2.636494 215 7 6.583441 653 3 3.282634 150 7 7.306894 [200 rows x 2 columns]	DATA2 Actual Predicted 4931 1426.1436 2367.484125 4148 1201.7690 2003.761745 7423 1836.2764 1495.824170 4836 2410.8618 1655.250543 944 1549.9824 2557.491625 ... ... 4644 3235.7880 2055.019906 6179 555.2772 1960.465134 1861 2885.5772 3566.508552 3598 218.3824 84.832269 1523 5478.2024 4634.783279 [1705 rows x 2 columns]	DATA3 Actual Predicted 7933 610 1018.735522 8657 1712 1677.327078 9599 642 928.630282 799 2018 1736.694061 3813 1776 1795.476231 ... ... 2890 829 1069.648481 4890 1539 1494.079432 2738 1817 1593.061998 5981 1658 1423.269266 6087 1626 1940.912959 [1999 rows x 2 columns]

Table (4.4) above shows the actual values and the values obtained from the SVR model. Where it is noted that the algorithm gives a better prediction with the first dataset compared to the second and third.

4.2.3 Multiple Linear Regression

The above Multiple Linear Regression algorithm gave the following results for each dataset separately:

Table 4.5: The Scores of Multiple Linear Regression with the Datasets.

Dataset 1	Dataset 2	Dataset 3
DATA1 Accuracy of our model is 82.87 score for Training: 0.8139340317312371 score for Testing: 0.8287043650324152 MAE for Training: 0.9306850912004401 MAE for Testing: 0.8384918299577447 MSE for Training: 1.6145583668446484 MSE for Testing: 1.3652262106916506 RMSE for Training: 1.2706527325924453 RMSE for Testing: 1.1684289497832765	DATA2 Accuracy of our model is 50.48 score for Training: 0.5012583266882864 score for Testing: 0.504814745913231 MAE for Training: 901.691461586371 MAE for Testing: 916.6681897190077 MSE for Training: 1450344.3865457668 MSE for Testing: 1449337.6418327559 RMSE for Training: 1204.3024481191453 RMSE for Testing: 1203.8843972046302	DATA3 Accuracy of our model is 38.13 score for Training: 0.36176597850517356 score for Testing: 0.38127321403488457 MAE for Training: 380.6605546032396 MAE for Testing: 373.858423672924 MSE for Training: 212663.5606747421 MSE for Testing: 206049.78313616617 RMSE for Training: 461.1545952007223 RMSE for Testing: 453.92706808050804

Table (4.5) above shows the scores obtained from the Multiple Linear Regression algorithm using various metrics and with the three datasets. Where it is noted that the first dataset with the highest feature correlation gave 82.87, followed by the second dataset with the medium correlation which gave 50.48, and then the third dataset with the weakest correlation which gave 38.13.

Table 4.6: The Actual and Predicted Values obtained by Multiple Linear Regression.

Dataset 1	Dataset 2	Dataset 3
DATA1 Actual Predicted 993 8 8.106782 859 6 5.804295 298 6 6.221682 553 4 4.493430 672 7 6.851528 .. .. 679 2 4.064490 722 3 2.632239 215 7 6.778160 653 3 3.322149 150 7 7.315458 [200 rows x 2 columns]	DATA2 Actual Predicted 4931 1426.1436 2374.217390 4148 1201.7690 2601.302740 7423 1836.2764 1743.899066 4836 2410.8618 1987.468583 944 1549.9824 2597.633847 ... ... 4644 3235.7880 2233.925147 6179 555.2772 1690.070707 1861 2885.5772 3634.620656 3598 218.3824 -302.814133 1523 5478.2024 5459.710602 [1705 rows x 2 columns]	DATA3 Actual Predicted 7933 610 1121.349350 8657 1712 1745.922558 9599 642 1040.170916 799 2018 1801.507622 3813 1776 1869.263796 ... ... 2890 829 1197.048999 4890 1539 1592.847815 2738 1817 1675.225921 5981 1658 1510.118113 6087 1626 1980.696687 [1999 rows x 2 columns]

Table (4.6) above shows the actual values and the values obtained from the Multiple Linear Regression model. Where it is noted that the algorithm gives a better prediction with the first dataset compared to the second and third.

4.2.4 Decision Tree

Decision Tree algorithm was applied to each dataset separately, as each dataset gave a different depth, and various results were obtained as well, as shown in the following tables:

Table 4.7 The Scores of the Decision Tree Algorithm with the Datasets.

Dataset 1	Dataset 2	Dataset 3
DATA1 Accuracy of our model is 96.29 score for Training: 0.9975387809877615 score for Testing: 0.96294304145097 MAE for Training: 0.04271368682645857 MAE for Testing: 0.2750800817539948 MSE for Training: 0.021356843413229284 MSE for Testing: 0.2953439596357689 RMSE for Training: 0.1461398077637619 RMSE for Testing: 0.5434555728261224	DATA2 Accuracy of our model is 58.28 score for Training: 0.6066630003240155 score for Testing: 0.5827708460774058 MAE for Training: 746.8756855024059 MAE for Testing: 780.1205776614123 MSE for Training: 1143826.8346673164 MSE for Testing: 1221171.093160397 RMSE for Training: 1069.4984033028363 RMSE for Testing: 1105.066103525213	DATA3 Accuracy of our model is 18.76 score for Training: 0.5722841265819558 score for Testing: 0.18762714677546144 MAE for Training: 274.9745634856751 MAE for Testing: 408.5757735012201 MSE for Training: 142517.5993989626 MSE for Testing: 270538.2311378747 RMSE for Training: 377.51503201721994 RMSE for Testing: 520.1328975731825

Table (4.7) above shows the scores obtained from the Decision Tree algorithm using various metrics and with the three datasets. Where it is noted that the first dataset with the highest feature correlation gave 96.29, followed by the second dataset with the medium correlation which gave 58.28, and then the third dataset with the weakest correlation which gave 18.76.

Table 4.8: The Actual and Predicted Values obtained by the Decision Tree.

Dataset 1	Dataset 2	Dataset 3
DATA1 Actual Predicted 993 8 8.0000 859 6 7.0000 298 6 5.0000 553 4 5.0000 672 7 7.0625 .. .. 679 2 2.0000 722 3 3.0000 215 7 7.0000 653 3 2.0000 150 7 7.0625 [200 rows x 2 columns]	DATA2 Actual Predicted 4931 1426.1436 2487.346175 4148 1201.7690 1714.752994 7423 1836.2764 2007.129594 4836 2410.8618 2007.129594 944 1549.9824 2964.155882 ... ... 4644 3235.7880 2007.129594 6179 555.2772 514.716399 1861 2885.5772 3975.682402 3598 218.3824 172.373913 1523 5478.2024 5923.966102 [1705 rows x 2 columns]	DATA3 Actual Predicted 7933 610 1304.881119 8657 1712 1818.234234 9599 642 673.333333 799 2018 1898.397590 3813 1776 1626.416667 ... ... 2890 829 1587.666667 4890 1539 1527.758065 2738 1817 1921.777778 5981 1658 1257.818182 6087 1626 1728.833333 [1999 rows x 2 columns]

Table (4.8) above shows the actual values and the values obtained from the Decision Tree algorithm. Where it is noted that the algorithm gives a better prediction with the first dataset compared to the second and third.

4.2.5 Random Forest

The random forest algorithm was applied to each dataset separately, as each dataset gave a different depth, and various results were obtained as well, as shown in the following tables:

Table 4.9: The Scores of the Random Forest Algorithm with the Datasets.

Dataset 1	Dataset 2	Dataset 3
DATA1 Accuracy of our model is 98.64 score for Training: 0.998254241434287 score for Testing: 0.9864120229419928 MAE for Training: 0.07766931177214641 MAE for Testing: 0.20858379646202244 MSE for Training: 0.01514854717919866 MSE for Testing: 0.1082961771523172 RMSE for Training: 0.12307943442833437 RMSE for Testing: 0.3290838451706756	DATA2 Accuracy of our model is 59.06 score for Training: 0.6129336215199575 score for Testing: 0.5905826293111134 MAE for Training: 741.7619127990797 MAE for Testing: 774.0955078981514 MSE for Training: 1125591.822960153 MSE for Testing: 1198307.1015592525 RMSE for Training: 1060.9391231169454 RMSE for Testing: 1094.6721434106435	DATA3 Accuracy of our model is 39.39 score for Training: 0.382169473172024 score for Testing: 0.39388732855127206 MAE for Training: 373.014536855395 MAE for Testing: 368.14008311645637 MSE for Training: 205864.98886577168 MSE for Testing: 201849.00240464805 RMSE for Training: 453.72347180388596 RMSE for Testing: 449.27608706078274

Table (4.9) above shows the scores obtained from the Random Forest algorithm using various metrics and with the three datasets. Where it is noted that the first dataset with the highest feature correlation gave 98.64, followed by the second dataset with the medium correlation which gave 59.06, and then the third dataset with the weakest correlation which gave 39.39.

Table 4.10: The Actual and Predicted Values obtained by the Random Forest.

Dataset 1	Dataset 2	Dataset 3
DATA1 Actual Predicted 993 8 7.880741 859 6 6.528804 298 6 6.031063 553 4 4.322625 672 7 7.050342 .. .. 679 2 2.430000 722 3 3.049763 215 7 7.061103 653 3 2.760323 150 7 7.050630 [200 rows x 2 columns]	DATA2 Actual Predicted 4931 1426.1436 2536.774174 4148 1201.7690 1862.279491 7423 1836.2764 2084.933578 4836 2410.8618 2200.374217 944 1549.9824 2992.466207 ... ... 4644 3235.7880 1927.380417 6179 555.2772 527.686875 1861 2885.5772 3999.428161 3598 218.3824 176.383063 1523 5478.2024 6252.451279 [1705 rows x 2 columns]	DATA3 Actual Predicted 7933 610 1241.076946 8657 1712 1782.677759 9599 642 919.778498 799 2018 1859.776887 3813 1776 1885.391518 ... ... 2890 829 1257.825030 4890 1539 1558.336603 2738 1817 1691.100327 5981 1658 1543.092047 6087 1626 2002.204360 [1999 rows x 2 columns]

Table (4.10) above shows the actual values and the values obtained from the Random Forest algorithm. Where it is noted that the algorithm gives a better prediction with the first dataset compared to the second and third.

4.2.6 K-Nearest Neighbors (KNN)

The above KNN Regression algorithm gave the following results for each dataset separately:

Table 4.11: The Scores of KNN with the Datasets.

Dataset 1	Dataset 2	Dataset 3
DATA1 Accuracy of our model is 50.40 score for Training: 0.7281483748987125 score for Testing: 0.5039899623588456 MAE for Training: 1.20625 MAE for Testing: 1.544 MSE for Training: 2.3589499999999997 MSE for Testing: 3.9532000000000007 RMSE for Training: 1.5358873656619485 RMSE for Testing: 1.9882655758223047	DATA2 Accuracy of our model is 18.39 score for Training: 0.4623196616598273 score for Testing: 0.18389034427633955 MAE for Training: 908.0393674743325 MAE for Testing: 1142.9383653020527 MSE for Training: 1563578.3055576135 MSE for Testing: 2388638.260411956 RMSE for Training: 1250.4312478331678 RMSE for Testing: 1545.522002564815	DATA3 Accuracy of our model is 28.46 score for Training: 0.4960598010584699 score for Testing: 0.2845602543794288 MAE for Training: 326.3733583489681 MAE for Testing: 392.392296148074 MSE for Training: 167916.02055784865 MSE for Testing: 238257.35005502752 RMSE for Training: 409.7755734031113 RMSE for Testing: 488.11612353519683

Table (4.11) above shows the scores obtained from the KNN algorithm using various metrics and with the three datasets. Where it is noted that the first dataset with the highest feature correlation gave 50.40, followed by the second dataset with the medium correlation which gave 18.39, and then the third dataset with the weakest correlation which gave 28.46.

Table 4.12: The Actual and Predicted Values obtained by KNN.

Dataset 1	Dataset 2	Dataset 3
DATA1 Actual Predicted 993 8 7.880741 859 6 6.528804 298 6 6.031063 553 4 4.322625 672 7 7.050342 .. .. 679 2 2.430000 722 3 3.049763 215 7 7.061103 653 3 2.760323 150 7 7.050630 [200 rows x 2 columns]	DATA2 Actual Predicted 4931 1426.1436 2536.774174 4148 1201.7690 1862.279491 7423 1836.2764 2084.933578 4836 2410.8618 2200.374217 944 1549.9824 2992.466207 ... ... 4644 3235.7880 1927.380417 6179 555.2772 527.686875 1861 2885.5772 3999.428161 3598 218.3824 176.383063 1523 5478.2024 6252.451279 [1705 rows x 2 columns]	DATA3 Actual Predicted 7933 610 1241.076946 8657 1712 1782.677759 9599 642 919.778498 799 2018 1859.776887 3813 1776 1885.391518 ... ... 2890 829 1257.825030 4890 1539 1558.336603 2738 1817 1691.100327 5981 1658 1543.092047 6087 1626 2002.204360 [1999 rows x 2 columns]

Table (4.12) above shows the actual values and the values obtained from the KNN. Where it is noted that the algorithm gives a better prediction with the first dataset compared to the second and third.

4.2.7 Bagging Regressor

The above Bagging Regressor gave the following results for each dataset separately:

Table 4.13: The Scores of Bagging Regressor with the Datasets.

Dataset 1	Dataset 2	Dataset 3
DATA1 Accuracy of our model is 98.35 score for Training: 0.9968005762131988 score for Testing: 0.983456712672522 MAE for Training: 0.09737499999999992 MAE for Testing: 0.22549999999999992 MSE for Training: 0.02776249999999999 MSE for Testing: 0.13185 RMSE for Training: 0.1666208270295163 RMSE for Testing: 0.36311155310730614	DATA2 Accuracy of our model is 51.69 score for Training: 0.9180418276590208 score for Testing: 0.5168584567188271 MAE for Training: 318.6691060369609 MAE for Testing: 838.9364282932552 MSE for Training: 238334.95684648244 MSE for Testing: 1414087.393014079 RMSE for Training: 488.19561330114635 RMSE for Testing: 1189.1540661386475	DATA3 Accuracy of our model is 30.82 score for Training: 0.8690650667262001 score for Testing: 0.3081999132933264 MAE for Training: 155.3285553470919 MAE for Testing: 381.696148074037 MSE for Training: 43628.337238273925 MSE for Testing: 230384.81777888944 RMSE for Training: 208.87397453554124 RMSE for Testing: 479.9841849258051

Table (4.13) above shows the scores obtained from the Bagging Regressor using various metrics and with the three datasets. Where it is noted that the first dataset with the highest feature correlation gave 98.35, followed by the second dataset with the medium correlation which gave 51.69, and then the third dataset with the weakest correlation which gave 30.82.

Table 4.14: The Actual and Predicted Values obtained by Bagging Regressor.

Dataset 1	Dataset 2	Dataset 3
DATA1 Actual Predicted 993 8 7.7 859 6 6.7 298 6 6.1 553 4 4.3 672 7 7.0 .. .. 679 2 2.4 722 3 3.1 215 7 7.0 653 3 2.8 150 7 7.0 [200 rows x 2 columns]	DATA2 Actual Predicted 4931 1426.1436 2560.73338 4148 1201.7690 2163.78342 7423 1836.2764 2026.49546 4836 2410.8618 1963.51078 944 1549.9824 1464.09420 ... ... 4644 3235.7880 1435.06532 6179 555.2772 559.13884 1861 2885.5772 6190.34208 3598 218.3824 161.98914 1523 5478.2024 6898.42038 [1705 rows x 2 columns]	DATA3 Actual Predicted 7933 610 1321.5 8657 1712 1870.4 9599 642 845.3 799 2018 1852.9 3813 1776 1722.1 ... ... 2890 829 1356.6 4890 1539 1473.2 2738 1817 1659.9 5981 1658 1645.3 6087 1626 1952.2 [1999 rows x 2 columns]

Table (4.14) above shows the actual values and the values obtained from the Bagging Regressor. Where it is noted that the algorithm gives a better prediction with the first dataset compared to the second and third.

4.2.8 Gaussian Processes Regressor

The above Gaussian Processes Regressor gave the following results for each dataset separately:

Table 4.15: The Scores of Gaussian Processes Regressor with the Datasets.

Dataset 1	Dataset 2	Dataset 3
DATA1 Accuracy of our model is -393.48 score for Training: 1.0 score for Testing: -3.934755332496863 MAE for Training: 5.487498766498789e-10 MAE for Testing: 5.6 MSE for Training: 3.878997604332376e-19 MSE for Testing: 39.33 RMSE for Training: 6.228159924353562e-10 RMSE for Testing: 6.271363488110062	DATA2 Accuracy of our model is -160.79 score for Training: 1.0 score for Testing: -1.6079167648232517 MAE for Training: 2.1644134214196608e-07 MAE for Testing: 2173.602127397745 MSE for Training: 7.579851271593981e-14 MSE for Testing: 7633005.835352951 RMSE for Training: 2.7531529691599015e-07 RMSE for Testing: 2762.789502541399	DATA3 Accuracy of our model is -574.12 score for Training: 1.0 score for Testing: -5.7411864881972345 MAE for Training: 1.7512772789169897e-07 MAE for Testing: 1368.067656229701 MSE for Training: 1.862104913749499e-12 MSE for Testing: 2244965.0564373797 RMSE for Training: 1.364589650315984e-06 RMSE for Testing: 1498.3207455139168

Table (4.15) above shows the scores obtained from the Gaussian Processes Regressor using various metrics and with the three datasets. It is noted that the prediction results of this algorithm in the training phase are excellent results up to 100%. But it gives opposite results in the testing phase, as the accuracy is negative.

Table 4.16: The Actual and Predicted Values obtained by Gaussian Processes Regressor.

Dataset 1	Dataset 2	Dataset 3
DATA1 Actual Predicted 993 8 0.000000e+00 859 6 8.473422e-308 298 6 1.979924e-191 553 4 0.000000e+00 672 7 0.000000e+00 .. .. 679 2 0.000000e+00 722 3 0.000000e+00 215 7 0.000000e+00 653 3 0.000000e+00 150 7 0.000000e+00 [200 rows x 2 columns]	DATA2 Actual Predicted 4931 1426.1436 1.180036e+02 4148 1201.7690 1.490493e-22 7423 1836.2764 2.551291e-02 4836 2410.8618 1.086757e-17 944 1549.9824 5.978265e-18 ... ... 4644 3235.7880 1.800219e-06 6179 555.2772 3.574022e+00 1861 2885.5772 1.718231e-32 3598 218.3824 6.415005e-01 1523 5478.2024 8.493056e-06 [1705 rows x 2 columns]	DATA3 Actual Predicted 7933 610 4.768511 8657 1712 1.786604 9599 642 26.953817 799 2018 0.037079 3813 1776 81.516149 ... ... 2890 829 85.818695 4890 1539 97.075038 2738 1817 3.722881 5981 1658 22.768672 6087 1626 0.001049 [1999 rows x 2 columns]

Table (4.16) above shows the actual values and the values obtained from the Gaussian Processes Regressor.As indicated above, the results obtained from this algorithm are inaccurate and negative.

4.2.9 RANSAC

The above RANSAC algorithm gave the following results for each dataset separately:

Table 4.17: The Scores of RANSAC with the Datasets.

Dataset 1	Dataset 2	Dataset 3
DATA1 Accuracy of our model is 83.10 score for Training: 0.8039736615464717 score for Testing: 0.8310306302685454 MAE for Training: 0.9395327263204999 MAE for Testing: 0.8240402474345082 MSE for Training: 1.700987922815109 MSE for Testing: 1.346685876759693 RMSE for Training: 1.304219277121416 RMSE for Testing: 1.160467955938333	DATA2 Accuracy of our model is 37.15 score for Training: 0.3691816554423569 score for Testing: 0.3715266146814903 MAE for Training: 1031.1911441917084 MAE for Testing: 1044.230269655462 MSE for Training: 1834424.3000272727 MSE for Testing: 1839453.268680271 RMSE for Training: 1354.409207007717 RMSE for Testing: 1356.2644538143256	DATA3 Accuracy of our model is 17.97 score for Training: 0.13510738556170498 score for Testing: 0.17972819080708424 MAE for Training: 404.35122146163286 MAE for Testing: 392.4079924370075 MSE for Training: 288187.61895040347 MSE for Testing: 273168.75918548054 RMSE for Training: 536.831089776294 RMSE for Testing: 522.65548804684

Table (4.17) above shows the scores obtained from the RANSAC using various metrics and with the three datasets. Where it is noted that the first dataset with the highest feature correlation gave 83.10, followed by the second dataset with the medium correlation which gave 37.15, and then the third dataset with the weakest correlation which gave 17.97.

Table 4.18: The Actual and Predicted Values obtained by RANSAC.

Dataset 1

DATA1

	Actual	Predicted
993	8	8.002545
859	6	6.064742
298	6	5.929001
553	4	4.824050
672	7	7.043432
..	...	...
679	2	4.585756
722	3	2.625224
215	7	6.764879
653	3	3.529801
150	7	7.160770

[200 rows x 2 columns]

Dataset 2

DATA2

	Actual	Predicted
4931	1426.1436	2302.632002
4148	1201.7690	3546.391834
7423	1836.2764	2101.696813
4836	2410.8618	2268.603175
944	1549.9824	3250.081898
...	...	...
4644	3235.7880	3446.453279
6179	555.2772	589.179296
1861	2885.5772	4373.202870
3598	218.3824	-521.403503
1523	5478.2024	5275.658852

[1705 rows x 2 columns]

Dataset 3

DATA3

	Actual	Predicted
7933	610	773.934205
8657	1712	1463.976109
9599	642	861.614376
799	2018	1505.147017
3813	1776	1784.230191
...	...	...
2890	829	738.482499
4890	1539	1409.698938
2738	1817	1495.239305
5981	1658	1098.991157
6087	1626	1854.740723

[1999 rows x 2 columns]

Table (4.18) above shows the actual values and the values obtained from the RANSAC. Where it is noted that the algorithm gives a better prediction with the first dataset compared to the second and third.

4.2.10 Gradient Boosting

The above Gradient Boosting algorithm gave the following results for each dataset separately:

Table 4.19: The Scores of Gradient Boosting with the Datasets.

Dataset 1	Dataset 2	Dataset 3
DATA1 Accuracy of our model is 98.65 score for Training: 0.9947820557792301 score for Testing: 0.9864506065483274 MAE for Training: 0.15137318282782958 MAE for Testing: 0.23548508218755138 MSE for Training: 0.04527789567194688 MSE for Testing: 0.10798866580983094 RMSE for Training: 0.21278603260540124 RMSE for Testing: 0.32861628962945666	DATA2 Accuracy of our model is 59.29 score for Training: 0.6361772511237832 score for Testing: 0.5928677649421101 MAE for Training: 723.417288782145 MAE for Testing: 774.0043644786283 MSE for Training: 1057999.18026998 MSE for Testing: 1191618.8307366394 RMSE for Training: 1028.5908711776417 RMSE for Testing: 1091.6129491429824	DATA3 Accuracy of our model is 39.29 score for Training: 0.4055721357210106 score for Testing: 0.39291736833004764 MAE for Training: 365.9422417209843 MAE for Testing: 367.6754785363584 MSE for Training: 198067.0755936456 MSE for Testing: 202172.0207348182 RMSE for Training: 445.0472734369301 RMSE for Testing: 449.6354309157789

Table (4.19) above shows the scores obtained from the Gradient Boosting using various metrics and with the three datasets. Where it is noted that the first dataset with the highest feature correlation gave 98.65, followed by the second dataset with the medium correlation which gave 59.29, and then the third dataset with the weakest correlation which gave 39.29.

Table 4.20: The Actual and Predicted Values obtained by Gradient Boosting.

Dataset 1	Dataset 2	Dataset 3
DATA1 Actual Predicted 993 8 7.984694 859 6 6.621468 298 6 6.269605 553 4 4.567035 672 7 7.362034 .. .. 679 2 2.548806 722 3 3.131077 215 7 6.941654 653 3 2.870800 150 7 7.025002 [200 rows x 2 columns]	DATA2 Actual Predicted 4931 1426.1436 2575.356148 4148 1201.7690 1823.046474 7423 1836.2764 2205.481699 4836 2410.8618 2308.559788 944 1549.9824 2570.141905 ... ... 4644 3235.7880 1633.075342 6179 555.2772 271.105896 1861 2885.5772 4219.868841 3598 218.3824 143.053232 1523 5478.2024 6344.436848 [1705 rows x 2 columns]	DATA3 Actual Predicted 7933 610 1254.682547 8657 1712 1788.652528 9599 642 865.623147 799 2018 1844.456012 3813 1776 1930.909621 ... ... 2890 829 1202.895468 4890 1539 1514.227424 2738 1817 1660.424720 5981 1658 1608.694695 6087 1626 2009.349664 [1999 rows x 2 columns]

Table (4.20) above shows the actual values and the values obtained from the Gradient Boosting. Where it is noted that the algorithm

gives a better prediction with the first dataset compared to the second and third.

4.2.11 Elastic Net

The above Elastic Net algorithm gave the following results for each dataset separately:

Table 4.21: The Scores of Elastic Net with the Datasets.

Dataset 1	Dataset 2	Dataset 3
DATA1 Accuracy of our model is 83.12 score for Training: 0.81253244674893 score for Testing: 0.8311804860328954 MAE for Training: 0.9269525184053404 MAE for Testing: 0.8252301272591549 MSE for Training: 1.6267204015309642 MSE for Testing: 1.3454915263178233 RMSE for Training: 1.275429496887603 RMSE for Testing: 1.1599532431601816	DATA2 Accuracy of our model is 47.67 score for Training: 0.47359067149266587 score for Testing: 0.4766855577721473 MAE for Training: 918.2627745702274 MAE for Testing: 935.9901887430127 MSE for Training: 1530802.127595154 MSE for Testing: 1531667.8220443104 RMSE for Training: 1237.255886062036 RMSE for Testing: 1237.6056811619403	DATA3 Accuracy of our model is 38.14 score for Training: 0.3617607936158005 score for Testing: 0.38139576447849877 MAE for Training: 380.6642074597479 MAE for Testing: 373.82418014195207 MSE for Training: 212665.28831225223 MSE for Testing: 206008.9711123412 RMSE for Training: 461.1564683621517 RMSE for Testing: 453.8821114698631

Table (4.21) above shows the scores obtained from the Elastic Net using various metrics and with the three datasets. Where it is noted that the first dataset with the highest feature correlation gave 83.12, followed by the second dataset with the medium correlation which gave 47.67, and then the third dataset with the weakest correlation which gave 38.14.

Table 4.22: The Actual and Predicted Values obtained by Elastic Net.

Dataset 1	Dataset 2	Dataset 3
DATA1 Actual Predicted 993 8 8.146357 859 6 5.839860 298 6 6.085745 553 4 4.462254 672 7 7.099549 .. .. 679 2 4.109229 722 3 2.655073 215 7 6.641164 653 3 3.432036 150 7 7.276554 [200 rows x 2 columns]	DATA2 Actual Predicted 4931 1426.1436 2507.690480 4148 1201.7690 1945.881464 7423 1836.2764 1498.013354 4836 2410.8618 1711.955612 944 1549.9824 2805.691201 ... ... 4644 3235.7880 2133.193825 6179 555.2772 2191.230956 1861 2885.5772 3879.586299 3598 218.3824 130.184602 1523 5478.2024 4784.375932 [1705 rows x 2 columns]	DATA3 Actual Predicted 7933 610 1123.800326 8657 1712 1747.737108 9599 642 1041.118613 799 2018 1803.285840 3813 1776 1866.790884 ... ... 2890 829 1196.242056 4890 1539 1590.889170 2738 1817 1674.731204 5981 1658 1510.747571 6087 1626 1983.316790 [1999 rows x 2 columns]

Table (4.22) above shows the actual values and the values obtained from the Elastic Net. Where it is noted that the algorithm gives a better prediction with the first dataset compared to the second and third.

4.2.12 Bayesian Ridge

The above Bayesian Ridge algorithm gave the following results for each dataset separately:

Table 4.23: The Scores of Bayesian Ridge with the Datasets.

Dataset 1	Dataset 2	Dataset 3
DATA1 Accuracy of our model is 82.98 score for Training: 0.8136326184672223 score for Testing: 0.8297984023813194 MAE for Training: 0.9282099904601893 MAE for Testing: 0.8316024434355436 MSE for Training: 1.6171738333473138 MSE for Testing: 1.3565067330208842 RMSE for Training: 1.271681498389952 RMSE for Testing: 1.1646916901141195	DATA2 Accuracy of our model is 50.47 score for Training: 0.5010502441964748 score for Testing: 0.5046923434138361 MAE for Training: 901.9176456815262 MAE for Testing: 916.8280269895121 MSE for Training: 1450949.4919341602 MSE for Testing: 1449695.8967451616 RMSE for Training: 1204.5536484250754 RMSE for Testing: 1204.0331792542768	DATA3 Accuracy of our model is 38.19 score for Training: 0.3616405261186926 score for Testing: 0.3819440087912205 MAE for Training: 380.6907452899664 MAE for Testing: 373.64978176356806 MSE for Training: 212705.36219315958 MSE for Testing: 205826.39356065865 RMSE for Training: 461.1999156473899 RMSE for Testing: 453.6809380618263

Table (4.23) above shows the scores obtained from the Bayesian Ridge using various metrics and with the three datasets. Where it is noted that the first dataset with the highest feature correlation gave 82.98, followed by the second dataset with the medium correlation which gave 50.47, and then the third dataset with the weakest correlation which gave 38.19.

Table 4.24: The Actual and Predicted Values obtained by Bayesian Ridge.

Dataset 1

DATA1

	Actual	Predicted
993	8	8.142148
859	6	5.830838
298	6	6.165648
553	4	4.477193
672	7	6.987599
..	...	...
679	2	4.085824
722	3	2.646448
215	7	6.693338
653	3	3.363717
150	7	7.290751

[200 rows x 2 columns]

Dataset 2

DATA2

	Actual	Predicted
4931	1426.1436	2389.179666
4148	1201.7690	2577.019627
7423	1836.2764	1716.146351
4836	2410.8618	1949.659590
944	1549.9824	2589.673797
...	...	...
4644	3235.7880	2264.139130
6179	555.2772	1753.812130
1861	2885.5772	3636.680020
3598	218.3824	-266.080328
1523	5478.2024	5438.840424

[1705 rows x 2 columns]

Dataset 3

DATA3

	Actual	Predicted
7933	610	1132.749164
8657	1712	1751.299818
9599	642	1048.414617
799	2018	1805.547362
3813	1776	1859.901957
...	...	...
2890	829	1189.625934
4890	1539	1584.492839
2738	1817	1668.724279
5981	1658	1511.678585
6087	1626	1991.498848

[1999 rows x 2 columns]

Table (4.24) above shows the actual values and the values obtained from the Bayesian Ridge. Where it is noted that the algorithm gives a better prediction with the first dataset compared to the second and third.

4.2.13 Kernel Ridge

The above Kernel Ridge algorithm gave the following results for each dataset separately:

Table 4.25: The Scores of Kernel Ridge with the Datasets.

Dataset 1	Dataset 2	Dataset 3
DATA1 Accuracy of our model is 70.30 score for Training: 0.7368733523042512 score for Testing: 0.7030264081912028 MAE for Training: 1.1647136096068424 MAE for Testing: 1.149701766226208 MSE for Training: 2.2832403718411576 MSE for Testing: 2.366879526716114 RMSE for Training: 1.5110395004238497 RMSE for Testing: 1.5384666154051292	DATA2 Accuracy of our model is 50.44 score for Training: 0.5011652200609962 score for Testing: 0.5044446924387913 MAE for Training: 901.8376083004017 MAE for Testing: 917.1237940126122 MSE for Training: 1450615.141290089 MSE for Testing: 1450420.7363424771 RMSE for Training: 1204.414854313118 RMSE for Testing: 1204.3341464653724	DATA3 Accuracy of our model is 26.24 score for Training: 0.2288213661825963 score for Testing: 0.2623657146103838 MAE for Training: 406.5847167866259 MAE for Testing: 396.07163295558297 MSE for Training: 256961.53551918032 MSE for Testing: 245648.62550964567 RMSE for Training: 506.91373577678905 RMSE for Testing: 495.62952445314

Table (4.25) above shows the scores obtained from the Kernel Ridge using various metrics and with the three datasets. Where it is noted that the first dataset with the highest feature correlation gave 70.30, followed by the second dataset with the medium correlation which gave 50.44, and then the third dataset with the weakest correlation which gave 26.24.

Table 4.26: The Actual and Predicted Values obtained by Kernel Ridge.

Dataset 1	Dataset 2	Dataset 3
DATA1 Actual Predicted 993 8 7.852913 859 6 5.681299 298 6 6.848683 553 4 5.332009 672 7 6.536616 .. .. 679 2 4.301696 722 3 3.242409 215 7 6.781291 653 3 3.211597 150 7 8.219479 [200 rows x 2 columns]	DATA2 Actual Predicted 4931 1426.1436 2373.433929 4148 1201.7690 2571.392975 7423 1836.2764 1748.446533 4836 2410.8618 1988.919128 944 1549.9824 2588.580719 ... ... 4644 3235.7880 2269.863617 6179 555.2772 1683.390472 1861 2885.5772 3628.180573 3598 218.3824 -316.521790 1523 5478.2024 5431.301605 [1705 rows x 2 columns]	DATA3 Actual Predicted 7933 610 1261.848549 8657 1712 2056.353274 9599 642 752.483628 799 2018 2306.819114 3813 1776 1660.824220 ... ... 2890 829 1085.163953 4890 1539 1225.414802 2738 1817 1777.339065 5981 1658 1710.919700 6087 1626 2347.188182 [1999 rows x 2 columns]

Table (4.26) above shows the actual values and the values obtained from the Kernel Ridge. Where it is noted that the algorithm gives a better prediction with the first dataset compared to the second and third.

Three tables will be listed below that represent the results achieved with each dataset separately. Table (4.27) below shows the results of ML algorithms with the first dataset:

Table 4.27: The Results of the Algorithms Using the First Dataset.

Algorithm	MSE	MAE	Accuracy (R^2)
Multilayer Perceptron	3.2698	1.3527	58.97
SVR	1.4116	0.8377	82.29
Multiple Linear Regression	1.3652	0.8384	82.87
Decision Tree	0.315	0.2750	96.05
Random Forest Regressor	0.1082	0.2085	98.64
K- Nearest Neighbor (KNN)	3.9532	1.544	50.40
Bagging Regressor	0.1318	0.2254	98.35
Gaussian Processes Regressor	39.33	5.6	-393.48
RANSAC	1.3466	0.8240	83.10
Gradient Boosting	0.1079	0.2354	98.65
Elastic Net	1.3454	0.8252	83.12
Bayesian Ridge	1.3565	0.8316	82.98
Kernel Ridge	2.3668	1.1497	70.30

Table (4.27) above summarizes the results of the algorithms applied to the first dataset. From these results, the best algorithm was obtained, which is (Gradient Boosting) with result (98.65). Table (4.28) below will show the results of the second dataset:

Table 4.28: The Results of the Algorithms Using the Second Dataset.

Algorithm	MSE	MAE	Accuracy (R^2)
Multilayer Perceptron	1545896.9	946.9159	47.18
SVR	1532939.8127	925.1295	47.63
Multiple Linear Regression	1449337.6418	916.6681	50.48
Decision Tree	1221171.0931	780.1205	58.28
Random Forest Regressor	1196933.5811	774.0955	59.11
K- Nearest Neighbor (KNN)	2388638.2604	1142.9383	18.39
Bagging Regressor	1469051.3589	838.9364	49.81
Gaussian Processes Regressor	7633005.8353	2173.6021	-160.79
RANSAC	1839453.2686	1044.2302	37.15
Gradient Boosting	1191712.2726	774.0043	59.29
Elastic Net	1531667.8220	935.9901	47.67
Bayesian Ridge	1449695.8967	916.8280	50.47
Kernel Ridge	1450420.7363	917.1237	50.44

Table (4.28) above lists the results of the algorithms applied to the second dataset with the medium correlation. It is noted that the obtained results are also medium. where the best algorithm is (Gradient Boosting) with results (59.28).

Table 4.29: The Results of the Algorithms Using the Third Dataset

Algorithm	MSE	MAE	Accuracy (R^2)
Multilayer Perceptron	214547.7335	375.1952	35.58
SVR	210174.2116	365.3286	36.89
Multiple Linear Regression	206049.7831	373.8584	38.13
Decision Tree	270538.2311	408.5757	18.76
Random Forest Regressor	201849.0024	368.1400	39.39

K- Nearest Neighbor (KNN)	238257.3500	392.3922	28.46
Bagging Regressor	230384.8177	381.6961	30.82
Gaussian Processes Regressor	2244965.0564	1368.0676	- 574.12
RANSAC	288187.6189	392.4079	17.97
Gradient Boosting	202172.0207	367.6754	39.29
Elastic Net	206008.9711	373.8241	38.14
Bayesian Ridge	205826.3935	373.6497	38.19
Kernel Ridge	245648.6255	396.0716	26.24

Table (4.29) above shows the results obtained from the third dataset with a weak correlation between its features. As noted, the results obtained are weak and inaccurate. And the best algorithm in terms of result is (Random Forest) with results (39.39).

For all of the above, it is concluded that the first dataset with a good correlation between its features gave the best result with the best ML algorithm was (Gradient Boosting). As for the second dataset with medium correlation, it gave average results, as the best algorithm in terms of accuracy with this dataset was (Gradient Boosting). As for the third dataset with weak correlation, it gave modest and weak result, as the best algorithm was (Random Forest). Finally, and based on this study, when using this model to predict the quantities of foodstuffs that can be sold, the degree of correlation of features must be measured, and then based on the degree, the best-suggested algorithm are used to obtain the best prediction accuracy.

4.3 Testing the Strategy

To test the proposed model, a real dataset of Iraqi Dates sales was used for the period (2002-2020). The details of this dataset were fully included in the third chapter. It contains several features, including types of Dates, their quantities, prices, number of palm trees, and year of production, as shown in Table (3.4). In the beginning, the dataset was read and pre-processed. Then, the degree of correlation between the features of the dataset was measured using the Correlation Matrix of the Seaborn library, as shown in Figure (4.1) below:

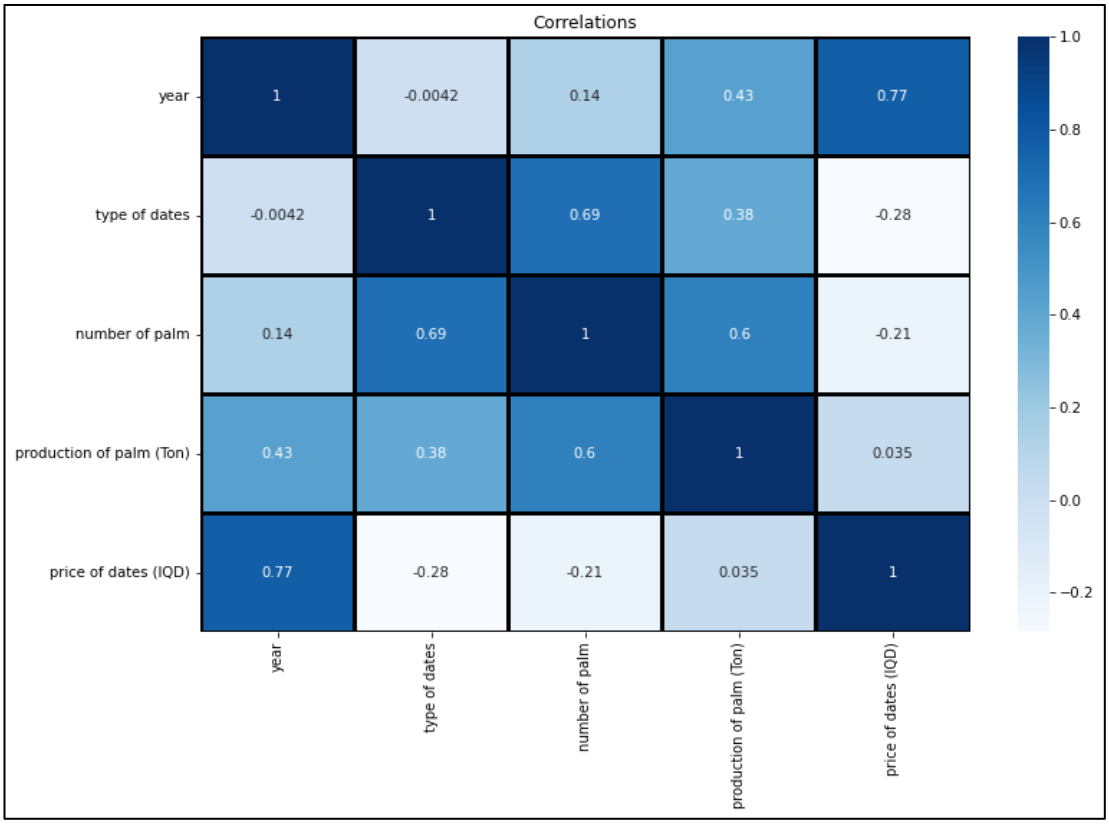


Figure 4.1:The correlation between Features of the Dataset.

Figure (4.1) above shows that the correlation of this dataset tends to be good. On this basis, the ML algorithm proposed by the model

previously was used, that worked better with datasets with good correlation, which are (Gradient Boosting). And when used, it gave prediction with excellent accuracy which is (99.51).

To predict the quantities of Dates that Iraq will sell for the next five years (2021-2025), Gradient Boosting algorithm was used and gave the result shown in part in Table (4.5) below:

Table 4.5: Prediction obtained from Gradient Boosting Algorithm.

	year	type_of_dates	number_of_palm	price_of_dates	production of palm (Ton)
0	2021	zahdi	7109731	500	394226.460527
1	2021	khestawy	1769092	650	82435.199339
2	2021	khadrawy	793020	861	32050.451504
3	2021	sair	1337643	866	27646.081264
4	2021	hellawy	665032	990	21764.191295
5	2021	others	5065321	1135	91519.700680
6	2022	zahdi	7279223	550	394226.460527
7	2022	khestawy	1878701	600	82435.199339
8	2022	khadrawy	797079	892	34414.046912
9	2022	sair	1366130	924	29612.478576
10	2022	hellawy	689599	906	21748.849035
11	2022	others	6089466	1241	354115.288511
12	2023	zahdi	7350548	730	393031.233921
13	2023	khestawy	1989611	500	82397.933673
14	2023	khadrawy	799917	908	34414.046912
15	2023	sair	1415111	995	28357.023022
16	2023	hellawy	697641	859	19545.927876
17	2023	others	6983229	1229	381679.448551
18	2024	zahdi	7604967	740	393031.233921
19	2024	khestawy	2120760	540	82397.933673
20	2024	khadrawy	810821	910	34414.046912
21	2024	sair	1455414	1037	42715.819884
22	2024	hellawy	701112	909	21748.849035
23	2024	others	7983429	1242	383704.883477
24	2025	zahdi	7429473	750	390194.142365
25	2025	khestawy	2259709	545	82397.933673

The above results can be used economically. Where the Gradient Boosting algorithm gave results with an accuracy of (99.51) and for five years (2021_2025). This model can be used with other datasets related to other crops or even sales related to various economic aspects. Where the specific dataset is used and the accuracy of the correlation between its features is measured, the artificial intelligence algorithm is used according to the degree of correlation to obtain the best prediction results.

CHAPTER FIVE

CONCLUSION AND FUTURE WORK

5.1 Overview

To find out the most important aspects of the study and to pave the way for subsequent studies, this chapter will deal with the conclusion and possible future works.

5.2 Conclusion

Restaurants and companies are making every effort to mitigate food waste and, consequently, increase profits. A model for predicting food sales by using artificial intelligence techniques is what this thesis proposed. The most important thing that hinders corporate profit is perishable products, and to avoid losing these products without selling them, this study focused on the quantities of products that can be sold later. The correlation between features of the datasets has a significant impact on the prediction results obtained from the ML algorithms. The model's prediction results are affected by the type of algorithm used, so determining the appropriate algorithm improves the results obtained.

5.3 Future Works

This study laid the foundation for the rest of the subsequent studies. It is possible to build on and expand this study in several aspects, the most important of which are:

1. Increasing the number of used datasets and diversifying them in terms of features and internal data.
2. Including more algorithms and diversify them to include most of the models used to predict sales of food products.

3. Using more metrics to measure the accuracy of the algorithms used to make the comparison between them more effective.
4. Using models that deal with time series. Time is an important Feature and these models work well in these cases.

REFERENCES

[1] M. Santalova, I. Soklakova, D. Balakhanova, E. Lesnikova, and E. Chudakova, "Target organizational structure and human potential," in *SHS Web of Conferences*, 2021, vol. 101: EDP Sciences, p. 02009.

[2] T. Chakraborty, S. S. Chauhan, and M. Ouhimmou, "Cost-sharing mechanism for product quality improvement in a supply chain under competition," *International Journal of Production Economics*, vol. 208, pp. 566-587, 2019.

[3] M. F. Ibrahim, Y. S. Mardhiyyah, A. Rusdiansyah, M. K. Boer, and D. M. Utama, "A three-phased perishable inventory simulation model with quality decrease consideration," *Jurnal Ilmiah Teknik Industri*, vol. 19, no. 2, pp. 198-211, 2020.

[4] B. Kuswandi, M. Moradi, and P. Ezati, "Food sensors: Off-package and on-package approaches," *Packaging Technology and Science*, vol. 35, no. 12, pp. 847-862, 2022.

[5] W. Wenji, "Study on inventions of fresh food in commercial aspects using e-commerce over internet," *Acta Agriculturae Scandinavica, Section B—Soil & Plant Science*, vol. 71, no. 4, pp. 303-310, 2021.

[6] B. Bilska, M. Tomaszewska, and D. Kołożyn-Krajewska, "The Management of Meals in Food Service Establishments in the Context of Food Waste—Results of Focus Group Interviews with Employees and Owners," *Sustainability*, vol. 14, no. 15, p. 9258, 2022.

[7] A. S. Sultan, M. A. Elgharib, T. Tavares, M. Jessri, and J. R. Basile, "The use of artificial intelligence, machine learning and deep learning in oncologic histopathology," *Journal of Oral Pathology & Medicine*, vol. 49, no. 9, pp. 849-856, 2020.

[8] A. Akella and S. Akella, "Machine learning algorithms for predicting coronary artery disease: efforts toward an open source solution," *Future Science OA*, vol. 7, no. 6, p. FSO698, 2021.

[9] S. Ravera *et al.*, "Discrete changes in glucose metabolism define aging," *Scientific reports*, vol. 9, no. 1, p. 10347, 2019.

[10] R. Ziols, "The Perceptron: A Partial History of Models and Minds in Data-Driven Educational Systems," in *The Educational Intelligent Economy: BIG DATA, Artificial Intelligence, Machine Learning and the Internet of Things in Education*: Emerald Publishing Limited, 2019.

[11] S. S. Raoof, M. Jabbar, and S. Tiwari, "Foundations of deep learning and its applications to health informatics," in *Deep Learning in Biomedical and Health Informatics*: CRC Press, 2021, pp. 1-28.

[12] I. A. Williamson, T. W. Hughes, M. Minkov, B. Bartlett, S. Pai, and S. Fan, "Reprogrammable electro-optic nonlinear activation functions for optical neural networks," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 26, no. 1, pp. 1-12, 2019.

[13] R. Gayathri, S. U. Rani, L. Čepová, M. Rajesh, and K. Kalita, "A Comparative Analysis of Machine Learning Models in Prediction of Mortar Compressive Strength," *Processes*, vol. 10, no. 7, p. 1387, 2022.

[14] S. Han, N. Mannan, D. C. Stein, K. R. Pattipati, and G. M. Bollas, "Classification and regression models of audio and vibration signals for

machine state monitoring in precision machining systems," *Journal of Manufacturing Systems*, vol. 61, pp. 45-53, 2021.
[15] G. Brindha, B. Rishiikeshwer, B. Santhi, K. Nakendraprasath, R. Manikandan, and A. H. Gandomi, "Precise prediction of multiple anticancer drug efficacy using multi target regression and support vector regression analysis," *Computer Methods and Programs in Biomedicine*, vol. 224, p. 107027, 2022.
[16] J. Manasa, R. Gupta, and N. Narahari, "Machine learning based predicting house prices using regression techniques," in *2020 2nd International conference on innovative mechanisms for industry applications (ICIMIA)*, 2020: IEEE, pp. 624-630.
[17] W. Xie, Y. She, and Q. Guo, "Research on multiple classification based on improved svm algorithm for balanced binary decision tree," *Scientific Programming*, vol. 2021, pp. 1-11, 2021.
[18] S. K. Dey *et al.*, "Prediction of dengue incidents using hospitalized patients, metrological and socio-economic data in Bangladesh: A machine learning approach," *PLoS One*, vol. 17, no. 7, p. e0270933, 2022.
[19] M. Awad and R. Khanna, *Efficient learning machines: theories, concepts, and applications for engineers and system designers*. Springer nature, 2015.
[20] S. Tripathi, A. Suryawanshi, and P. Selvaraju, "Can stock prices be predicted? Comparative study of LSTM and SVR for financial market forecast," 2022.
[21] M. Piekutowska *et al.*, "The application of multiple linear regression and artificial neural network models for yield prediction of very early potato cultivars before harvest," *Agronomy*, vol. 11, no. 5, p. 885, 2021.
[22] M. Sinurat, M. Heikal, A. Simanjuntak, R. Siahaan, and R. N. Ilham, "Product Quality On Consumer Purchase Interest With Customer Satisfaction As A Variable Intervening In Black Online Store High Click Market: Case Study on Customers of the Tebing Tinggi Black Market Online Store," *Morfai Journal*, vol. 1, no. 1, pp. 13-21, 2021.
[23] D. Alita, A. D. Putra, and D. Darwis, "Analysis of classic assumption test and multiple linear regression coefficient test for employee structural office recommendation," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 15, no. 3, pp. 1-5, 2021.
[24] M. Ahmad, N. A. Al-Shayea, X.-W. Tang, A. Jamal, H. M. Al-Ahmadi, and F. Ahmad, "Predicting the pillar stability of underground mines with random trees and C4. 5 decision trees," *Applied Sciences*, vol. 10, no. 18, p. 6486, 2020.
[25] S. Tangirala, "Evaluating the impact of GINI index and information gain on classification using decision tree classifier algorithm," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 2, pp. 612-619, 2020.
[26] C. Peláez-Rodríguez, J. Pérez-Aracil, D. Fister, L. Prieto-Godino, R. Deo, and S. Salcedo-Sanz, "A hierarchical classification/regression algorithm for improving extreme wind speed events prediction," *Renewable Energy*, vol. 201, pp. 157-178, 2022.
[27] D. Cannizzaro, A. Aliberti, L. Bottaccioli, E. Macii, A. Acquaviva, and E. Patti, "Solar radiation forecasting based on convolutional neural network and ensemble learning," *Expert Systems with Applications*, vol. 181, p. 115167, 2021.

[28] X. Qiu, L. Zhang, P. N. Suganthan, and G. A. Amaratunga, "Oblique random forest ensemble via least square estimation for time series forecasting," *Information Sciences,* vol. 420, pp. 249-262, 2017.
[29] M. Rakhra *et al.*, "Crop price prediction using random forest and decision tree regression:-a review," *Materials Today: Proceedings,* 2021.
[30] Q. Zhang, "Financial data anomaly detection method based on decision tree and random forest algorithm," *Journal of Mathematics,* vol. 2022, 2022.
[31] S. Desai and T. B. Ouarda, "Regional hydrological frequency analysis at ungauged sites with random forest regression," *Journal of Hydrology,* vol. 594, p. 125861, 2021.
[32] S. Bishnoi and H. Hisar, "k–NEAREST NEIGHBOR (k-NN) ALGORITHM FOR CLASSIFICATION," *Dr. Med Ram Verma,* p. 109.
[33] D. Lopez-Bernal, D. Balderas, P. Ponce, and A. Molina, "Education 4.0: teaching the basics of KNN, LDA and simple perceptron algorithms for binary classification problems," *Future Internet,* vol. 13, no. 8, p. 193, 2021.
[34] L. Kuang, H. Yan, Y. Zhu, S. Tu, and X. Fan, "Predicting duration of traffic accidents based on cost-sensitive Bayesian network and weighted K-nearest neighbor," *Journal of Intelligent Transportation Systems,* vol. 23, no. 2, pp. 161-174, 2019.
[35] B. Wang *et al.*, "A novel weighted KNN algorithm based on RSS similarity and position distance for Wi-Fi fingerprint positioning," *IEEE Access,* vol. 8, pp. 30591-30602, 2020.
[36] H. ŞAHİN and İ. Duygu, "Application of Random Forest Algorithm for the Prediction of Online Food Delivery Service Delay," *Turkish Journal of Forecasting,* vol. 5, no. 1, pp. 1-11.
[37] M. Shahhosseini, G. Hu, and H. Pham, "Optimizing ensemble weights and hyperparameters of machine learning models for regression problems," *Machine Learning with Applications,* vol. 7, p. 100251, 2022.
[38] T. N. Rincy and R. Gupta, "Ensemble learning techniques and its efficiency in machine learning: A survey," in *2nd International Conference on Data, Engineering and Applications (IDEA),* 2020: IEEE, pp. 1-6.
[39] Y. C. Yabansu, A. Iskakov, A. Kapustina, S. Rajagopalan, and S. R. Kalidindi, "Application of Gaussian process regression models for capturing the evolution of microstructure statistics in aging of nickel-based superalloys," *Acta Materialia,* vol. 178, pp. 45-58, 2019.
[40] D. V. Dao *et al.*, "A sensitivity and robustness analysis of GPR and ANN for high-performance concrete compressive strength prediction using a Monte Carlo simulation," *Sustainability,* vol. 12, no. 3, p. 830, 2020.
[41] J. Wang, "An intuitive tutorial to Gaussian processes regression," *arXiv preprint arXiv:2009.10862,* 2020.
[42] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.
[43] H. Jin *et al.*, "State-of-health estimation for lithium-ion batteries with hierarchical feature construction and auto-configurable Gaussian process regression," *Energy,* vol. 262, p. 125503, 2023.
[44] C. Lara-Alvarez and F. Gonzalez-Herrera, "Testing multiple polynomial models for eye-tracker calibration," *Behavior Research Methods,* vol. 52, no. 6, pp. 2506-2514, 2020.

[45] M. Ashphaq, P. K. Srivastava, and D. Mitra, "Analysis of univariate linear, robust-linear, and non-linear machine learning algorithms for satellite-derived bathymetry in complex coastal terrain," *Regional Studies in Marine Science,* vol. 56, p. 102678, 2022.
[46] J. M. Martínez-Otzeta, I. Rodríguez-Moreno, I. Mendialdua, and B. Sierra, "RANSAC for Robotic Applications: A Survey," *Sensors,* vol. 23, no. 1, p. 327, 2022.
[47] J. Li, Q. Hu, and M. Ai, "Robust geometric model estimation based on scaled Welsch q-norm," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 58, no. 8, pp. 5908-5921, 2020.
[48] H. K. Sangappa and K. Ramakrishnan, "A probabilistic analysis of a common RANSAC heuristic," *Machine Vision and Applications,* vol. 30, pp. 71-89, 2019.
[49] A. S. Ponraj and T. Vigneswaran, "Daily evapotranspiration prediction using gradient boost regression model for irrigation planning," *The Journal of Supercomputing,* vol. 76, pp. 5732-5744, 2020.
[50] S. Fafalios, P. Charonyktakis, and I. Tsamardinos, "Gradient Boosting Trees," *Gnosis Data Analysis PC,* pp. 1-3, 2020.
[51] Y. Zhang and A. Haghani, "A gradient boosting method to improve travel time prediction," *Transportation Research Part C: Emerging Technologies,* vol. 58, pp. 308-324, 2015.
[52] W. Liu, H. Fan, and M. Xia, "Step-wise multi-grained augmented gradient boosting decision trees for credit scoring," *Engineering Applications of Artificial Intelligence,* vol. 97, p. 104036, 2021.
[53] E. Spiliotis, "Decision trees for time-series forecasting," *Foresight,* vol. 1, pp. 30-44, 2022.
[54] B. Zhang, J. Ren, Y. Cheng, B. Wang, and Z. Wei, "Health data driven on continuous blood pressure prediction based on gradient boosting decision tree algorithm," *IEEE Access,* vol. 7, pp. 32423-32433, 2019.
[55] M. Y. Shams, O. M. Elzeki, L. M. Abouelmagd, A. E. Hassanien, M. Abd Elfattah, and H. Salem, "HANA: a healthy artificial nutrition analysis model during COVID-19 pandemic," *Computers in Biology and Medicine,* vol. 135, p. 104606, 2021.
[56] B. Yu, C. Chen, X. Wang, Z. Yu, A. Ma, and B. Liu, "Prediction of protein–protein interactions based on elastic net and deep forest," *Expert Systems with Applications,* vol. 176, p. 114876, 2021.
[57] R. Garcia-Carretero *et al.*, "Logistic LASSO and elastic net to characterize vitamin D deficiency in a hypertensive obese population," *Metabolic syndrome and related disorders,* vol. 18, no. 2, pp. 79-85, 2020.
[58] M. Roozbeh, M. Maanavi, and N. A. Mohamed, "Penalized least squares optimization problem for high-dimensional data," *International Journal of Nonlinear Analysis and Applications,* vol. 14, no. 1, pp. 245-250, 2023.
[59] Q. Xu, X. Ding, C. Jiang, K. Yu, and L. Shi, "An elastic-net penalized expectile regression with applications," *Journal of Applied Statistics,* vol. 48, no. 12, pp. 2205-2230, 2021.
[60] R. McElreath, *Statistical rethinking: A Bayesian course with examples in R and Stan*. Chapman and Hall/CRC, 2020.

[61] P. Arora, D. Boyne, J. J. Slater, A. Gupta, D. R. Brenner, and M. J. Druzdzel, "Bayesian networks for risk prediction using real-world data: a tool for precision medicine," *Value in Health,* vol. 22, no. 4, pp. 439-445, 2019.
[62] Y. Jin, W. Fu, J. Kang, J. Guo, and J. Guo, "Bayesian symbolic regression," *arXiv preprint arXiv:1910.08892*, 2019.
[63] R. van de Schoot *et al.*, "Bayesian statistics and modelling," *Nature Reviews Methods Primers,* vol. 1, no. 1, p. 1, 2021.
[64] Y. YANG and Y. YANG, "Hybrid Prediction Method for Wind Speed Combining Ensemble Empirical Mode Decomposition and Bayesian Ridge Regression."
[65] I. Mihaylov, M. Nisheva, and D. Vassilev, "Application of machine learning models for survival prognosis in breast cancer studies," *Information,* vol. 10, no. 3, p. 93, 2019.
[66] A. Shahsavar, M. Jamei, and M. Karbasi, "Experimental evaluation and development of predictive models for rheological behavior of aqueous Fe3O4 ferrofluid in the presence of an external magnetic field by introducing a novel grid optimization based-Kernel ridge regression supported by sensitivity analysis," *Powder Technology,* vol. 393, pp. 1-11, 2021.
[67] P. N. Suganthan, "On non-iterative learning algorithms with closed-form solution," *Applied Soft Computing,* vol. 70, pp. 1078-1082, 2018.
[68] H. Mai, T. C. Le, D. Chen, D. A. Winkler, and R. A. Caruso, "Machine learning for electrocatalyst and photocatalyst design and discovery," *Chemical Reviews,* vol. 122, no. 16, pp. 13478-13515, 2022.
[69] T. Kusumoto, K. Mitarai, K. Fujii, M. Kitagawa, and M. Negoro, "Experimental quantum kernel trick with nuclear spins in a solid," *npj Quantum Information,* vol. 7, no. 1, p. 94, 2021.
[70] M. Kapralov, N. Nouri, I. Razenshteyn, A. Velingker, and A. Zandieh, "Scaling up Kernel Ridge Regression via Locality Sensitive Hashing," *arXiv preprint arXiv:2003.09756,* 2020.
[71] X. Wang, D. Lin, W. Fan, and T. Wang, "Research on Sales Forecast of Fresh Produce Considering Weather Factors," 2018.
[72] V. Prabhakar, D. Sayiner, U. Chakraborty, T. Nguyen, and M. Lanham, "Demand Forecasting for a large grocery chain in Ecuador," *Data. Published,* 2018.
[73] Y. F. Akande, J. Idowu, A. Misra, S. Misra, O. N. Akande, and R. Ahuja, "Application of XGBoost Algorithm for Sales Forecasting Using Walmart Dataset," in *Advances in Electrical and Computer Technologies: Select Proceedings of ICAECT 2021*: Springer, 2022, pp. 147-159.
[74] Y. Liu, "Grocery Sales Forecasting," in *2022 International Conference on Creative Industry and Knowledge Economy (CIKE 2022)*, 2022: Atlantis Press, pp. 215-219.
[75] A. Schmidt, M. W. U. Kabir, and M. T. Hoque, "Machine Learning Based Restaurant Sales Forecasting," *Machine Learning and Knowledge Extraction,* vol. 4, no. 1, pp. 105-130, 2022.
[76] T. S. Yange, C. O. Egbunu, O. Onyekwere, and K. A. Foga, "Prediction of Agro Products Sales Using Regression Algorithm," *American Journal of Data Mining and Knowledge Discovery*, vol. 5, no. 1, pp. 11-19, 2020.

[77] T. Tanizaki, T. Hoshino, T. Shimmura, and T. Takenaka, "Demand forecasting in restaurants using machine learning and statistical analysis," *Procedia CIRP*, vol. 79, pp. 679-683, 2019.
[78] T. Weng, W. Liu, and J. Xiao, "Supply chain sales forecasting based on lightGBM and LSTM combination model," *Industrial Management & Data Systems*, vol. 120, no. 2, pp. 265-279, 2020.
[79] I. Vallés-Pérez, E. Soria-Olivas, M. Martínez-Sober, A. J. Serrano-López, J. Gómez-Sanchís, and F. Mateo, "Approaching sales forecasting using recurrent neural networks and transformers," *Expert Systems with Applications*, vol. 201, p. 116993, 2022.
[80] X. Bi, G. Adomavicius, W. Li, and A. Qu, "Improving Sales Forecasting Accuracy: A Tensor Factorization Approach with Demand Awareness," *INFORMS Journal on Computing*, vol. 34, no. 3, pp. 1644-1660, 2022.
[81] N. Kumar, K. Dheenadayalan, S. Reddy, and S. Kulkarni, "Multimodal Neural Network For Demand Forecasting," *arXiv e-prints*, p. arXiv: 2210.11502, 2022.
[82] H. Dai, H. Yu, Q. Xiao, and W. Zhou, "A Clustering-based Sales Forecast Method for Big Promotion Days in O2O On-Demand Retailing," in *2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2019: IEEE, pp. 1079-1083.
[83] C.-H. Wang, "Considering economic indicators and dynamic channel interactions to conduct sales forecasting for retail sectors," *Computers & Industrial Engineering*, vol. 165, p. 107965, 2022.
[84] R. Odegua, "Applied Machine Learning for Supermarket Sales Prediction," *Project: Predictive Machine Learning in Industry*, 2020.
[85] A. P. Wellens, R. N. Boute, and M. Udenio, "A Tree-based Framework to Democratize Large-scale Retail Sales Forecasting with Big Data," *Available at SSRN*, 2022.
[86] J. Wolters and A. Huchzermeier, "Joint in-season and out-of-season promotion demand forecasting in a retail environment," *Journal of Retailing*, vol. 97, no. 4, pp. 726-745, 2021.
[87] N. Vairagade, D. Logofatu, F. Leon, and F. Muharemi, "Demand forecasting using random forest and artificial neural network for supply chain management," in *Computational Collective Intelligence: 11th International Conference, ICCCI 2019, Hendaye, France, September 4–6, 2019, Proceedings, Part I 11*, 2019: Springer, pp. 328-339.
[88] V. Adithya Ganesan, S. Divi, N. B. Moudhgalya, U. Sriharsha, and V. Vijayaraghavan, "Forecasting food sales in a multiplex using dynamic artificial neural networks," in *Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC), Volume 2*, 2019: Springer, pp. 69-80.
[89] L. Zhou, C. Zhang, F. Liu, Z. Qiu, and Y. He, "Application of deep learning in food: a review," *Comprehensive reviews in food science and food safety*, vol. 18, no. 6, pp. 1793-1811, 2019.
[90] S. Nosratabadi, S. Ardabili, Z. Lakner, C. Mako, and A. Mosavi, "Prediction of food production using machine learning algorithms of multilayer perceptron and ANFIS," *Agriculture*, vol. 11, no. 5, p. 408, 2021.

[91] R. Valbuena *et al.*, "Evaluating observed versus predicted forest biomass: R-squared, index of agreement or maximal information coefficient?," *European Journal of Remote Sensing*, vol. 52, no. 1, pp. 345-358, 2019.
[92] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Computer Science*, vol. 7, p. e623, 2021.
[93] F. Rustam *et al.*, "COVID-19 future forecasting using supervised machine learning models," *IEEE access*, vol. 8, pp. 101489-101499, 2020.
[94] M. H. D. M. Ribeiro, R. G. da Silva, V. C. Mariani, and L. dos Santos Coelho, "Short-term forecasting COVID-19 cumulative confirmed cases: Perspectives for Brazil," *Chaos, Solitons & Fractals*, vol. 135, p. 109853, 2020.
[95] D. Alita, A. D. Putra, and D. Darwis, "Analysis of classic assumption test and multiple linear regression coefficient test for employee structural office recommendation."

الخلاصة

في الوقت الحاضر، يعتمد اقتصاد الدول على تنمية القطاع الخاص. ومن أهم الشركات العاملة في هذا القطاع هي شركات المواد الغذائية. وقد عانت هذه الشركات من مشكلة تخزين المنتجات وتلفها بسبب تاريخ انتهاء صلاحيتها. تقترح هذه الأطروحة بناء استراتيجية تتنبأ بالحاجة الفعلية للشركات لكميات المنتجات في ظل وجود متغيرات أخرى. تعتمد الاستراتيجية المقترحة على ركيزتين أساسيتين: الأولى هي استخدام ثلاث مجموعات بيانات غذائية مختلفة مع اختلاف الارتباط بين سماتها، حيث أن مجموعة البيانات الأولى ذات ارتباط مرتفع، والثانية ذات ارتباط متوسط، والثالثة ستكون ضعيفة الارتباط. والثانية هي استخدام ثلاثة عشر خوارزمية للتعلم الآلي وتقييم نتائجها بناءً على عدة مقاييس محددة للحصول على أفضل خوارزمية من حيث الدقة. تشير النتائج المتحصل عليها إلى أن أفضل خوارزمية مطبقة على مجموعة البيانات الأولى ذات الارتباط العالي هي Gradient Boosting التي أعطت دقة (98.65)، وأفضل خوارزمية مطبقة على مجموعة البيانات الثانية ذات الارتباط المتوسط هي Gradient Boosting ايضا والتي أعطت دقة (59.29)، في حين أن أفضل خوارزمية مطبقة على مجموعة البيانات الثالثة مع الارتباط المنخفض كانت Random Forest التي أعطت دقة (39.39). وبناء على هذه النتائج تم تطبيق النموذج المقترح للتنبؤ بكميات التمور التي سينتجها العراق للسنوات الخمس القادمة، مع توفر متغيرات أخرى. حيث كانت مجموعة البيانات ذات ارتباط جيد، تم استخدام الخوارزمية الأولى وأعطت دقة (99.51). تضع هذه الأطروحة حداً للهدر الغذائي اليومي الذي يكبد العالم خسائر مالية كبيرة.

جامعة كربلاء
كلية علوم الحاسوب وتكنولوجيا المعلومات
قسم علوم الحاسوب

التنبؤ بمبيعات الأغذية باستخدام تقنيات التعلم الآلي

رسالة ماجستير
مقدمة الى مجلس كلية علوم الحاسوب وتكنولوجيا المعلومات / جامعة كربلاء وهي جزء من متطلبات نيل درجة الماجستير في علوم الحاسوب

كتبت بواسطة
حسام مزهر مرداس بريسم

بإشـــراف
أ.م.د اياد حميد موسى

١٤٤٤ هـ

٢٠٢٣ م