REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
MACHINE FACULTY
DEPARTMENT OF INDUSTRIAL ENGINEERING

INVENTORY AGE PREDICTION USING DEEP LEARNING ALGORITHMS IN ALCHOLIC BEVERAGE INDUSTRY

MELİS KAMACIOĞLU **18061034**

ÖZGÜR GÜMÜŞ **19061062**

UNDERGRADUATE THESIS

ADVISOR
PROF. DR. ALEV TAŞKIN

ISTANBUL, 2023

PREFACE

We sincerely thank to our thesis advisor Prof. Dr. Alev TAŞKIN for her motivation, support and time she spent throughout out thesis. Also, we are grateful to our families for their support and love.

June, 2023

Melis Kamacıoğlu, Özgür Gümüş

CONTENTS

LIST OF SYMBOLS

Y	Dependent Variable
X	Independent Variable
β0	Intercept of the line
β1	Linear regression coefficient (slope of the line)
ε	Random error
Wij	Weight in the first layer, from input i to hidden unit j
wj0	Bias for hidden unit j.
g(.)	Activation funciton
z0	Extra hidden unit
k	Output unit
f	Sigmoid function
Vji	Weights between input layer and hidden layer
Wkj	Weights between hidden layer and output layer
ft	Forget gate
it	Input gate
ot	Output gate
c	Memory cell
Wx	Internal recurrent connections
Win	Input-to-hidden layer connections
ϕ(·)	Non-linear function
ρ	Spectral radius
n	The number of data points
y_i	The actual (observed) value of the dependent variable
$\hat{y}_i$	The predicted value of the dependent variable
Σ	Summation symbol

LIST OF ABBREVIATIONS

ESN	Echo State Networks
MLP	Multilayer Perceptron
RNN	Recurrent Neural Networks
CNN	Convolutional Neural Network
KNN	K-Nearest Neighbors
SVM	Support Vector Machine
LSTM	Long-Short Term Memory
FMCG	Fast Moving Consumer Goods
XGBoost	Extreme Gradient Boosting
IoT	Internet of Things
SLR	Simple Linear Regression
OLS	Ordinary Least Squares
RBF	Radial Basis Function
ANN	Artificial Neural Network
ReLU	Rectified Linear Unit
BPTT	Backpropagation Through Time
MSE	Mean Square Error
MSD	Mean Square Deviation
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
ESS	Explained Sum of Squares
TSS	Total Sum of Squares
SGD	Stochastic Gradient Descent

LIST OF FIGURES

Page

LIST OF TABLES

ÖZET

ALKOLLÜ İÇECEK ENDÜSTRİSİNDE DERİN ÖĞRENME ALGORİTMALARI KULLANARAK STOK YAŞI TAHMİNİ

Melis Kamacıoğlu, Özgür Gümüş

Endüstri Mühendisliği Bölümü

Lisans Bitirme Tezi

Tez Danışmanı: Prof. Dr. Alev Taşkın

Bu tez, Doğu Avrupa ve Türkiye'de faaliyet gösteren bir alkollü içecek firmasının depolarındaki stok yaşının tahminine odaklanmaktadır. Temel amaç, özellikle alkol gibi çabuk bozulan mallar için, depodaki ürün depolama süresini doğru bir şekilde tahmin ederek, toplam maliyetlerin önemli bir kısmı olan depolama maliyetlerini en aza indirmektir, bozulma ve kayıpları önlemek için stok yaşı çok önemlidir. Bunu yaparak, bu tez içerisinde, şirketin tedarik zinciri operasyonlarının verimliliğini ve etkinliğini artırmayı amaçlamaktadır. Bu tezin katkıları arasında titiz veri toplama süreci, öznitelik seçimi ve korelasyon matrisi kullanılarak değerlendirme, önemli özniteliklerin istatistiksel analizi, üç adet makine öğrenmesi modelinin oluşturulması ve dört adet derin öğrenme algoritmasının modeli yer alıyor. Araştırma, bu tahmin tekniklerinin performansını karşılaştırmaktadır. Modellerde gözlemlenen sonuçlar ve eksiklikler, stok yaşını tahmin etmek için etkili bir yöntem olarak ESN'nin en uygun olduğunu söylemektedir.

Stok yaşı tahmini, talep tahmini ve optimum envanter yönetimine odaklanan çoğu çalışma ile karşılaştırıldığında daha az araştırılan bir konudur. Hareketli ortalama, zaman serisi tahmini ve üstel düzeltme gibi teknikler yaygın olarak kullanılırken, derin öğrenme ve makine öğrenmesi yöntemlerine daha az rastlanır.

Bu tez, hızlı tüketim malları endüstrisinde stok yaşı tahmini için içgörüler ve tavsiyeler sağlayarak depo yönetimi ve maliyet düşürme gibi konularda potansiyel faydalar sunar.

Son olarak, bu alanda daha fazla araştırma yapılması gerektiğinin altını çizen bu tez, gelecekteki çalışmalar için ise ESN algoritmalarının kullanılmasını önermektedir.

Anahtar Kelimeler: Stok yaşı, envanter yönetimi, tahmin, makine öğrenmesi, derin öğrenme, yapay sinir ağları, alkollü içecek endüstrisi, echo state networks.

ABSTRACT

INVENTORY AGE PREDICTION USING DEEP LEARNING ALGORITHMS IN ALCHOLIC BEVERAGE INDUSTRY

Melis Kamacıoğlu, Özgür Gümüş

Department of Industrial Engineering

BSc. Thesis

Adviser: Prof. Dr. Alev Taşkın

This thesis focuses on the prediction of inventory age in warehouses for an alcoholic beverage company operating in Eastern Europe and Turkey. The main objective is to minimize storage costs, a significant portion of the total costs, by accurately predicting the duration of product storage in the warehouse, specifically for perishable goods like alcohol, inventory age is crucial to prevent spoilage and losses. By doing so, the thesis aims to enhance the efficiency and effectiveness of the company's supply chain operations. The contributions of this thesis include the rigorous data collection process, feature selection and evaluation using correlation matrix, statistical analysis of important values, construction of three machine learning models, and modeling of four deep learning algorithms. The research compares the performance of these prediction techniques. The results and deficiencies observed in the models inform the recommendation of ESN as an effective method for predicting inventory age.

Inventory age prediction is a less explored topic, with most studies focusing on demand forecasting and optimal inventory management. Common techniques such as moving average, time series forecasting, and exponential smoothing are commonly used, while deep learning and machine learning methods are less prevalent.

This thesis provides insights and recommendations for inventory age prediction in the FMCG industry, offering potential benefits for warehouse management and cost reduction. It also highlights the need for further research in this area and suggests the utilization of ESN algorithms for future studies.

Keywords: Inventory age, inventory management, prediction, machine learning, deep learning, neural networks, alcoholic beverage industry, echo state networks.

CHAPTER 1

INTRODUCTION

This thesis researches about inventory age prediction on warehouses for a alcoholic beverage company which manages its operations on Eastern Europe and Turkey. The purpose of this thesis is to minimize the storage cost, which has a significant share in the total cost, by predicting the storage duration of the products in the warehouse. In this way, it is aimed to increase the efficiency and effectiveness of the supply chain operations of the company. As a result of our research, it was claimed that the best method for this prediction study would be Echo State Network (ESN) and the results were compared by applying machine learning, deep learning and neural network techniques, which are frequently used in prediction.

1.1 Literature Review

Inventory Age prediction is not frequently mentioned topic in literature. In recent years, there has been increase of attention about demand forecasting, ordering optimal quantity prediction, optimal storage level in inventory management topic. In this review it has been seen that deep and machine learning methods are used rarely. Mostly used techniques are moving average, time series forecasting, simple and quadratic exponantial smoothing and trend analysis.

In the literature, there are numerous amount of articles about inventory management. Altough, number of articles about inventory management prediction are rare.

Correlation matrix is used for feature selection because it provides insights into the relationships between different features in a dataset. It measures the statistical dependence between pairs of variables, indicating how changes in one variable are associated with changes in another variable. Correlation Matrix is commonly used for feature selection in many studies. In Usman and Evan's research in 2022 [2] and Yuling and Weiqin's research in 2022 [3] shows that it is a sufficient technique for optimal feature selection.

Linear Regression, Polynomial Regression, Ridge Regression and Support Vector Machine's are kinds of machine learning algortihms used for prediction of a spesific data. In the literature, This algorithms is mostly used for prediction problems. This problems varies as such cases like wind speed detection and fire prediction [4][5]. Multiple linear regression has been commonly used in the scientific community,

especially in the COVID-19 studies [6]. This models is not sufficient to every prediction problems but it is a helpful tool to be able to analyze the complexity of the problem.

According to Mohammed and Nqishbandi in 2023 [7], MLP is a fundamental deep learning architecture composed of multiple layers of interconnected artificial neurons. With its ability to model complex relationships between variables, MLP has been applied to mental health prediction tasks by utilizing historical data as input features in their work. Brenes and Johannssenn in 2022 [8] have demonstrated that MLP models can capture intricate patterns and nonlinearity in predicting the bankruptcy prediction models

In 2023, Ünal and Erkayman [9] observed 59 articles and came to the conclusion of machine learning algorithms are the most commonly used for inventory management. They inspected LSTM, RNN, CNN, XGBoost, KNN. During the research, 8 clusters are created such as inventory maganement, inventory optimization, demand forecasting, ABC analysis, supply chain magament, classification, machine learning, forecasting. The clusters are a result of selected research or studies which is about inventory management between 2012 and 2022 years. This article guides to find most suitable algorithms for inventory management topic and to better understand the literature.

Min Han Kim and Yong Su Kim in 2010 [12], used RNNs are specialized deep learning architectures designed to handle sequential data for prediction of indoor air quality problem. In the context of air quality prediction, RNNs can effectively model the temporal dependencies present in air pollulant, considering factors such as product demand and sales trends over time. By exploiting the recurrent nature of RNNs, researchers have achieved notable success in accurately predicting indoor air quality based on air pollutants data that are collected in a subway station.

Deng and Liu in 2021 [10], propose LSTM model which is a type of recurrent neural network (RNN) architecture that is widely used for modeling sequential data. It addresses the vanishing gradient problem in traditional RNNs and can effectively capture long-term dependencies in time series data. LSTM models have been successfully applied. By considering historical order data, LSTM can learn complex temporal patterns and accurately forecast. The ability of LSTM to capture dependencies over long sequences makes it particularly suitable for predicting demand, which often relies on historical trends and patterns.

Selvin and Vinayakumar in 2017 [11], have conducted comparative analyses of LSTM and RNN for stock price prediction in stock price data. These analyses focus on evaluating the performance and accuracy of both algorithms. Researchers have found that LSTM and RNN can achieve similar prediction accuracies, with slight variations depending on the specific dataset and features. The choice between LSTM and RNN depends on factors such as dataset size, computational resources, and the specific requirements of the stock price prediction system. In conclusion, RNN has better results for this model.

Hao Zhang in 2021 [13], focuses on the application of an ESN prediction model in the stock market using compressed sensing. The authors propose a method that combines ESNs with compressed sensing techniques to forecast stock market behavior. Compressed sensing is used to reduce the dimensionality of the input data, making it

more efficient for ESNs to process. The authors demonstrate the effectiveness of their approach by conducting experiments on real-world stock market data.

Li Liu in 2023 [14] investigates the use of ESNs for predictive control of energy-saving in central heating systems. They propose an ESN-based model that optimizes energy consumption in central heating systems by predicting future demand. The ESN model is trained using historical data, and the predictions are used to adjust the heating system settings to minimize energy consumption while maintaining comfort levels. The authors conduct experiments to evaluate the performance of their model and compare it with traditional control strategies.

According to Cesar H. Valencia in 2023 [15], Reservoir selection plays a crucial role in the performance of ESNs, and the authors propose a method to automatically select an optimal reservoir configuration based on the characteristics of the time series data. They also propose a hyperparameter optimization model to tune the parameters of the ESNs for improved forecasting accuracy. The authors evaluate their approach on various time series datasets and demonstrate its effectiveness compared to traditional ESN models.

Sujan Ghimire in 2023 [16], seeks a novel approach that integrates Convolutional Neural Networks (CNNs) with ESNs for daily electricity demand prediction. The authors propose a hybrid model that combines the strengths of CNNs in capturing spatial dependencies in the electricity demand data with the temporal modeling capabilities of ESNs. The CNNs are used to extract features from the input data, which are then fed into the ESNs for prediction. The authors evaluate their approach using real-world electricity demand data and show that the hybrid model outperforms standalone CNN and ESN models in terms of prediction accuracy.

Jei-Zheng Wu in 2013 [17], has adressed the practical management of operations according to inventory aging processes. The aim of this work is to construct a polynomial-time based model to acquire important information, such as inventory ages, accounting principles, and product structures, for the correct forecast of inventory write-downs to lessen the impact of the carrying value fluctuation of inventory. Research indicates that the presented model presupposes linearity and deterministic parameters and variables, and that simulation and evolutionary methods may be used to create more broad models. The examination of the consequences of supply chain decisions and changes to accounting rules will be made possible by this expansion.

1.2 Problem Definition

The amount of days that pass until a corporation sells its inventory balance is called the age of inventory. Analysts use this metric to evaluate the efficacy of sales. Also, Investors can assess a company's performance using the age of its inventory. The age of the inventory provides information on how quickly a company turns over its stock. Moreover, it is a crucial measure that managers should also consider. Managers can determine the appropriateness of their pricing approach by keeping an eye on the age of their inventory. If their products inventory age is lower than that of other businesses, the corporation might be overcharging for its goods. However, if their age

of inventory is higher than that of comparable businesses, the company may be overcharging for its goods and failing to move enough inventory. Additionally, aging inventory can affect decisions made when developing marketing strategies, such as giving discounts and promotions, selling aged inventory, and improving cash flow. Inventory is subject to obsolescence risk if its average age exceeds a certain threshold. For our problem, the age of inventory is very important in the fast moving consumer goods (FMCG) industries, especially for perishable food and beverage goods that has expiration date, such as alcohol. The management of businesses who sell these products must pay particular attention to the average age of their inventory because if it spoils, it must be written off completely and could incur significant losses.

In the firm there is a 3 big warehouses within the factories that are alcohol is produced. There are 62 products and 19 product groups for the company. The alcohol company checks the inventory age for each product to be able to imply inventory management techniques for all warehouses. In some cases, products can take longer time to leave the warehouse for such as logistic problems, lower order quantities, lack of resources, order cancellations and political and economic regulations. To be able to manage this problems need of new and innovative methods of inventory management foreseen. With the help of this new methods, company can take actions like, marketing campaings, better scheduling for production planning, optimized logistic routes that can prevent holding unnecessary stocks and reduce costs.

1.3 Hypothesis

The inventory age problem is a critical problem for companies working in the FMCG field that produces perishable products. In this research, necessary data were collected and evaluations were made to provide a clearer understanding of the problem. Although there is not much research in this area, the existing researches are mostly focused on inventory management. For this reason, in this research, it has been tried to get a better result by using the common methods of machine and deep learning, as well as the Echo State Network (ESN) method, which offers a new and uncommon perspective.

CHAPTER 2

INVENTROY MANAGEMENT AND PREDICTION

Inventory is the supply of raw materials, partially finished goods called work-in-progress and finished goods, an organization maintains to meet its operational needs. It represents a sizeable investment and a potential source of waste that needs to be carefully controlled. Inventory is defined as a stock of goods that is maintained by a business in anticipation of some future demand [18]. It is one of the most important company assets because it has a direct effect on the company's ability to obtain opinions [19].

Inventory management plays a critical role in the success of businesses across various industries. Efficient inventory management ensures that the right products are available in the right quantities at the right time, optimizing customer satisfaction, reducing carrying costs, and enhancing profitability. However, accurately predicting future inventory needs can be a complex task, influenced by numerous factors such as seasonality, demand patterns, supplier lead times, and economic conditions. In this section, we discuss the importance of inventory management prediction and explore the techniques and approaches used to forecast future inventory levels.

2.1 Importance of Inventory Management and Prediction

The importance of inventory management is underscored by its role in the success of any business, irrespective of size or industry. It involves the oversight of the flow of goods from the point of procurement to the point of sale or use. Inventory management is crucial for several reasons. First and foremost, it ensures that customer demand is consistently met, preventing stockouts and lost sales, which ultimately enhances customer satisfaction and loyalty. Moreover, it enables businesses to control costs by optimizing inventory levels and reducing carrying costs like storage and obsolescence. By accurately tracking inventory and demand patterns, businesses can streamline production planning, minimize disruptions, and improve overall efficiency. Additionally, efficient inventory management contributes to better supply chain management, resulting in reduced lead times and enhanced collaboration with suppliers and distributors. Proper management of inventory also helps businesses manage their cash flow effectively by freeing up working capital and minimizing excess

inventory. It ensures accurate financial reporting, as inventory valuation directly impacts financial statements. Finally, by minimizing waste and obsolescence through monitoring and replenishment strategies, businesses can reduce financial losses.

Effective inventory management prediction offers numerous advantages for businesses operating in dynamic market environments. By accurately forecasting demand, companies can reduce the risk of stockouts and overstock situations, leading to improved customer satisfaction and increased profitability [20].

Accurate prediction of inventory requirements is essential for achieving optimal inventory levels, avoiding stockouts, and minimizing excess inventory. Traditional inventory management approaches often rely on historical sales data and basic forecasting methods, which may not adequately capture the complexities and uncertainties inherent in demand patterns. As businesses face increasing competition, shorter product lifecycles, and fluctuating customer demands, the need for more sophisticated inventory management prediction techniques becomes apparent.

By leveraging advanced predictive analytics and machine learning algorithms, businesses can gain valuable insights into future demand patterns, enabling proactive inventory planning and replenishment strategies. Effective inventory management prediction can lead to improved customer satisfaction through enhanced product availability, reduced costs through optimized inventory levels, and increased operational efficiency through streamlined supply chain processes.

2.2 Techniques for Inventory Management and Prediction

A variety of techniques and approaches have been developed to predict inventory needs accurately. These methods range from traditional statistical models to more advanced machine learning algorithms. These techniques leverage historical sales data, market trends, external factors, and other relevant variables to forecast future demand patterns [21]. Some commonly used techniques include time series analysis, machine learning algorithms (e.g., random forests, support vector machines), and artificial neural networks [22]. These models capture the underlying patterns and relationships within the data, enabling the generation of reliable demand forecasts [23].

2.2.1 Machine Learning Algorithms

Machine learning algorithms offer powerful tools for inventory management prediction by leveraging complex patterns and relationships present in large datasets. Techniques such as regression models, random forests, and neural networks can be trained on historical sales data along with other relevant factors like promotions, weather conditions, and economic indicators. These algorithms learn from the data and generate forecasts that can capture nonlinear relationships and dynamic demand patterns, improving the accuracy of inventory predictions.

2.2.2 Deep Learning Algorithms

Deep learning techniques give increased capabilities for inventory management prediction by using complicated patterns and linkages identified in large datasets. Convolutional neural networks (CNNs), recurrent neural networks (RNNs), and deep neural networks (DNNs) are some of the techniques that can be used to train deep learning models on historical sales data as well as a variety of contextual factors. These algorithms learn from the data and provide predictions that are excellent at capturing non-linear correlations and dynamic demand patterns, improving the accuracy of forecasting the requirement for inventory. Businesses may optimize their inventory management methods and make educated judgments to satisfy client demand while reducing costs by utilizing the depth and complexity of deep learning.

2.2.3 Demand Sensing and Real-Time Data Analytics

In addition to historical data, demand sensing and real-time data analytics techniques provide valuable insights into current market conditions and immediate demand signals. By integrating data from various sources such as point-of-sale systems, customer sentiment analysis, and social media monitoring, businesses can detect shifts in demand patterns and respond promptly to changing customer preferences. These techniques enable adaptive inventory management, allowing businesses to adjust their inventory levels and supply chain operations in real-time, enhancing responsiveness and reducing stockouts or overstocks.

2.3 Challenges and Future Directions

While inventory management prediction techniques have made significant advancements, challenges persist in achieving highly accurate and robust forecasts. Issues such as data quality, demand volatility, supply chain disruptions, and the need for timely and reliable data integration remain critical areas for further research and improvement. Future directions in inventory management prediction involve the exploration of advanced machine learning algorithms, hybrid forecasting approaches that combine multiple techniques, and the integration of emerging technologies like Internet of Things (IoT) devices and big data analytics.

Efficient inventory management prediction has the potential to revolutionize supply chain operations, leading to improved customer satisfaction, cost savings, and enhanced competitive advantage. By addressing the challenges and exploring new avenues for improvement, businesses can harness the power of predictive analytics to optimize their inventory management practices and thrive in dynamic market environments.

Finally, inventory management prediction offers significant opportunities for businesses to optimize their supply chain operations. By leveraging advanced predictive modeling techniques and robust data analytics, companies can achieve accurate demand forecasts, optimize inventory levels, and improve resource allocation. The benefits of inventory management prediction extend beyond cost

reduction, encompassing enhanced customer satisfaction, improved operational efficiency, and increased profitability. However, organizations must address challenges related to data availability, data quality, and market dynamics to fully harness the potential of predictive analytics in inventory management.

CHAPTER 3

METHODOLOGY

In this thesis, an model was built with the inventory age predicting problem. In this section, explanations of the methodology are given. The main steps of this thesis can be listed as: methodolgy, data collection, identifying the problem, data evaluation, literature review, outlier elimination, correlation matrix results evaluation, finding suitable algorithms to apply, building models, comparison of methods and results and discussion. The flow diagram of the methodology is given in Figure 3.1.

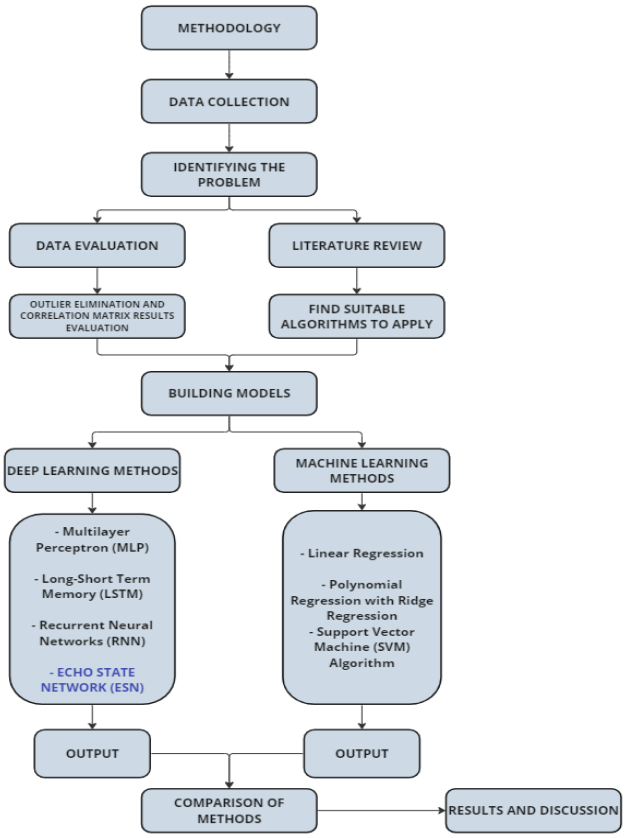

Fig 3. 1 Work Flow Chart

3.1 Prediction Problems

Insufficient or Low-Quality Data: The availability of relevant and reliable data is crucial for making predictions. Unreliable or misleading forecasts may result from inadequate, incorrect, or biased data utilized in the prediction process.

Outliers and Anomalies: Predictions may be affected by outliers, which are data points that differ considerably from the norm. Outliers can distort forecasts or produce incorrect results if they are not appropriately managed or discovered.

Models for predicting inventory age must be resistant to outliers and anomalies that might bias forecasts. Anomalies in the data can be caused by unanticipated occurrences like sudden shifts in customer behavior, supply interruptions, or regulatory changes. It is essential to create models that can manage and take into consideration such anomalies.

Overfitting or Underfitting: An overfitted prediction model has a poor ability to generalize to new data because it is overly complicated and begins to collect noise or random fluctuations in the training data. On the other side, underfitting occurs when a model is overly straightforward and fails to recognize significant patterns in the data, resulting in poor predicted performance.

Changing or Dynamic Patterns: Predictions based on historical data could not be accurate in the future if the underlying patterns or correlations between the variables change over time. This is especially important when working with complicated systems or situations that are changing quickly.

Confounding Variables: Confounding variables are factors that are correlated with both the input variables and the predicted outcome. Failure to account for confounding variables can lead to biased predictions and incorrect conclusions.

Lack of Causal Understanding: Instead than focusing on causes, predictive models frequently emphasize correlations. Even while forecasts based on correlations may be helpful, they might not offer an in-depth understanding of the underlying responsible mechanisms, which may restrict the capacity to make precise predictions.

Uncertainty and Probability: Predictions are by their very nature unpredictable, particularly when they include complicated systems and few data. Understanding the accuracy and constraints of predictions requires communicating and measuring uncertainty.

Limited data availability: Finding sufficient, high-quality historical data on inventory age can be difficult, particularly if the sector or the particular company has insufficient data gathering procedures. Predictions made with little data may turn out to be incorrect or unreliable. In the context of inventory age prediction, the variables impacting inventory age, such as the product type, brand popularity, marketing activities, and price variations, should be identified and included in the characteristics. However, gathering thorough and trustworthy feature data might be difficult.

Seasonal and temporal variations: Seasonal variations, public holidays, and special occasions can all affect the demand for alcoholic drinks. It can be difficult to capture and include these fluctuations in the prediction models since they frequently call for

taking into consideration a variety of variables, including the weather, cultural events, and consumer preferences.

Supply chain complexities: Complex supply chains with several stakeholders, including as suppliers, distributors, and retailers, are a feature of the alcoholic beverage business. It is necessary to take into account a number of variables, such as manufacturing lead times, transportation delays, and stock replenishment patterns, in order to accurately anticipate inventory age. It might be difficult to include these complex supply chain dynamics in the prediction models.

Generalization across products: There are many different types of spirits, wines, and beers included in the large range of goods made by the alcoholic beverage business. Predictive models have to be able to generalize across many product categories and correctly forecast the age in each type of inventory. A big problem might be ensuring model scalability and responsiveness to various product attributes.

Model interpretability: It can be challenging to evaluate and comprehend the reasoning behind the predictions made by deep learning algorithms like neural networks since they frequently function as "black boxes." Developing models that enable interpretability, enabling stakeholders to comprehend and trust the predictions generated, might be essential in sectors like the alcoholic beverage business, where regulatory compliance and transparency are critical.

3.2 Machine Learning Methods

3.2.1 Simple Linear Regression

The simplest linear regression model finds the relationship between one input variable, which is called the predictor variable, and the output, which is called the system's response. This type of model is known as a simple linear regression (SLR). [24] Regression is a method for simulating an objective value using separate predictors. Linear Regression is an algorithm that part of supervised Machine Learning. The main applications of this technique are forecasting and determining the causal connections between variables. The number of independent variables and the nature of the relationship between the independent and dependent variables are the main determinants of how regression algorithms differ. A type of regression analysis known as simple linear regression involves only one independent variable and a linear connection between the independent (x) and dependent (y) variables. The best-fit straight line is indicated by the line in the graph above. We attempt to draw a line from the provided data points that best represents the points. The linear equation below can be used to represent the line.

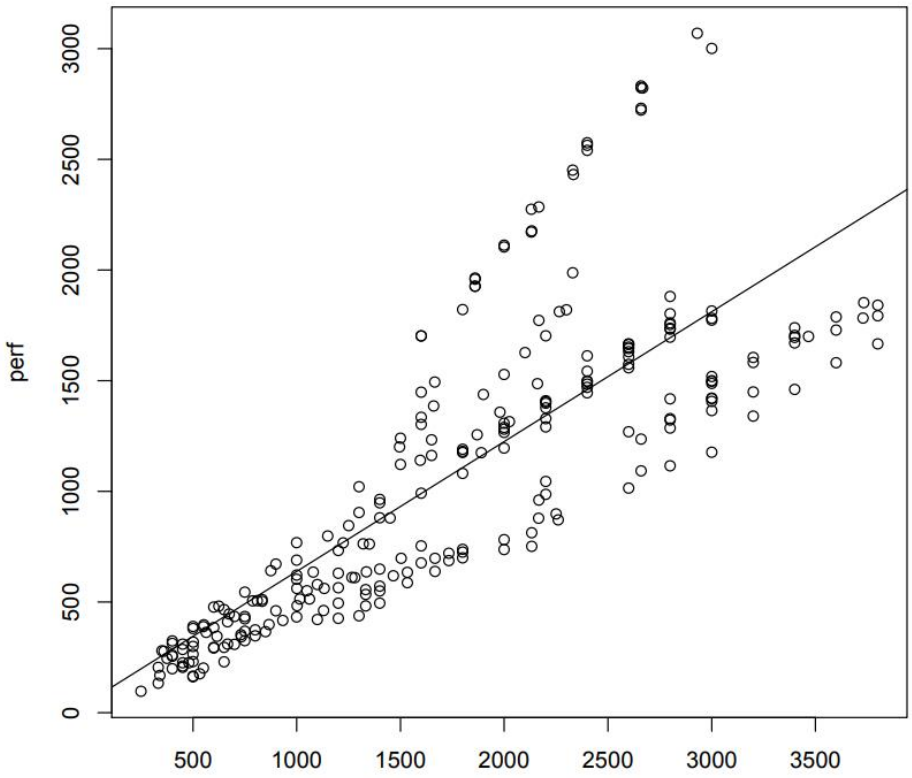

Fig 3. 2 Linear Regression Graph

Mathematical representations of linear regression are as follows:

$$y= \beta 0+ \beta\ 1x+ \varepsilon \quad (3.1)$$

Y= Dependent Variable

X= Independent Variable

β0= Intercept of the line

β1 = Linear regression coefficient (slope of the line)

ε = Random error

The last parameter, random error ε, is required as the best fit line also doesn't include the data points perfectly. Finding the best values for β0 and β1 is the goal of the linear regression procedure.

Linear Regression Types

There are two different types of linear regression algorithms:

Simple Linear Regression is a type of linear regression method used to predict the value of a numerical dependent variable using only one independent variable. This regression is used into this thesis.

Multiple Linear Regression is the name given to a Linear Regression technique in which more than one independent variable is utilized to forecast the value of a numerical dependent variable.

Cost Function: The optimal values for β0 and β1 that would produce the best fit line for the data points can be determined using the cost function. We convert the finding problem into a minimization problem because we want to minimize the difference between the predicted value and the actual value since we want the best values for β0 and β1 [25].

$$minimize \frac{1}{n} \sum_{i=1}^{n} (pred_i - y_i)^2 \tag{3.2}$$

$$J = \frac{1}{n} \sum_{i=1}^{n} (pred_i - y_i)^2 \tag{3.3}$$

3.2.2 Polynomial Regression

In statistics, polynomial regression is a form of linear regression in which the relationship between the independent variable x and the dependent variable y is modeled as an nth order polynomial. [26] Simply said, it is a development of linear regression where the connection between the variables is not purely linear but can be better illustrated by a curved line [27]. More complicated interactions between the variables can be captured by choosing a higher degree for the polynomial. It's crucial to remember that adopting a very high degree can result in overfitting, when the model fits the training data excessively closely but fails to perform well on new, unexpected information. Polynomial regression fits a nonlinear relationship between the value of x and the corresponding conditional mean of y, denoted as $\mathrm{E}(y/x)$. Although polynomial fits a non linear model to the data, as statistical estimation problem it is linear, in the sense that the regression function $\mathrm{E}(y/x)$ is linear in the unknown parameters that are estimated from the data [26].

The following is the equation for polynomial regression:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \ldots + \beta_k x_i^k + e_i \quad i = 1,2,\ldots n \tag{3.4}$$

y= Dependent Variable

X= Independent Variable

β0= Intercept of the line

βi = Linear regression coefficient (slope of the line)

ε = Random error

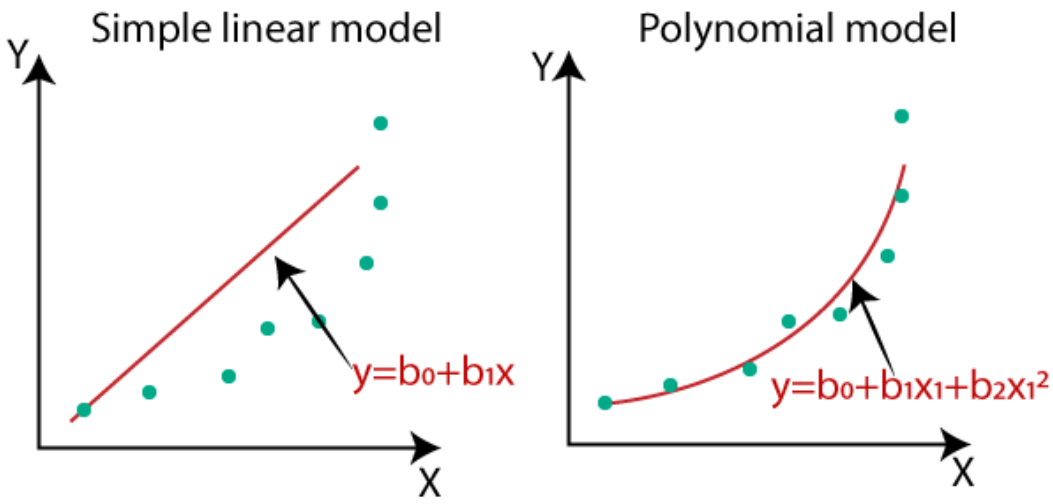

Fig 3. 3 Difference of Simple Linear Regression and Polynomial Regression

3.2.3 Ridge Regression

Ridge regression is a statistical approach for modeling the relationship between a dependent variable and one or more independent variables in regression analysis. It is a regularized form of linear regression that addresses problems like multicollinearity and overfitting by adding a penalty term to the ordinary least squares (OLS) approach.

In ridge regression, the OLS equation's sum of squared residuals is increased by the penalty term, also referred to as the ridge term. The squared value of the independent variable coefficients is directly proportional to this term. Ridge regression places a restriction on the coefficients, forcing them to decrease in size until they reach zero. The regularization parameter, often known as lambda (λ), is a tuning parameter that regulates the amount of shrinkage.

In ridge regression, the OLS equation's sum of squared residuals is increased by the penalty term, also referred to as the ridge term. The squared value of the independent

variable coefficients is directly proportional to this term. Ridge regression provides a restriction on the coefficients, forcing them to decrease in size until they reach zero. The regularization parameter, often known as lambda (λ), is a tuning parameter that regulates the amount of shrinkage. As can be seen in the Fig 3. 4, increasing the parameters in the model increases the complexity of the model. When the complexity of the model increases, high variance has low bias. Where complexity is low, high bias has low variance [28]. Bias, variance need to be prioritized and understood to avoid overfit.

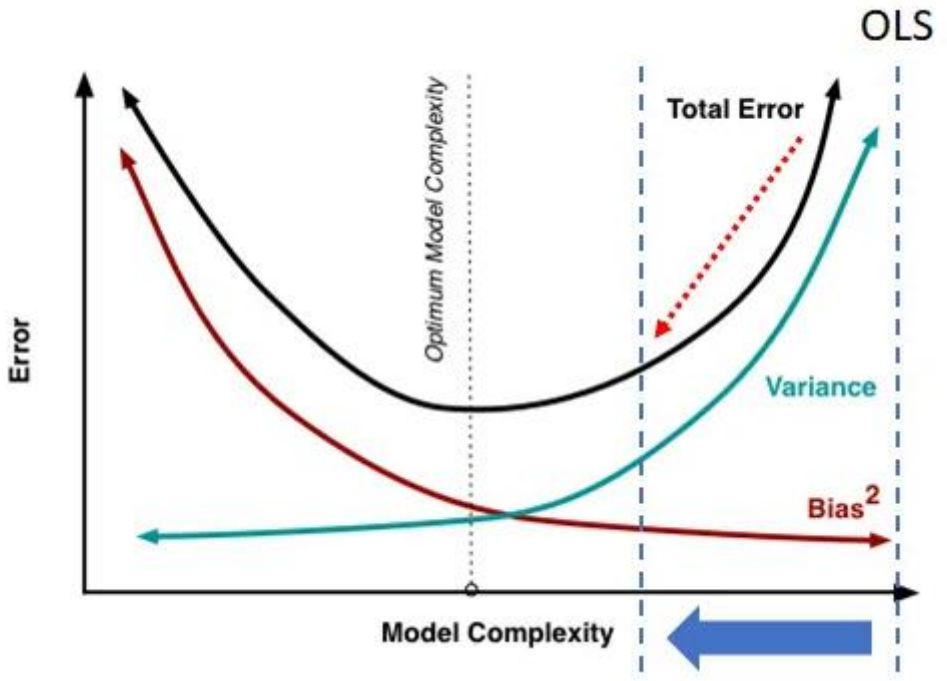

Fig 3. 4 OLS Chart

Ridge regression may be beneficial in avoiding overfitting, which happens when a model matches the training data too closely and performs poorly on unobserved data. Ridge regression discourages big coefficients by include the ridge component, resulting in a more straightforward and reliable model.

When dealing with datasets that suffer from the "curse of dimensionality," or a high ratio of features to observations, ridge regression is particularly helpful. It is frequently employed in a variety of disciplines, including as finance, economics, and machine learning, where multicollinearity or overfitting are issues with linear regression.

3.2.4 Support Vector Machine (SVM)

The supervised learning technique known as SVM, or Support Vector Machine, can be utilized for classification or regression tasks. The SVM algorithmic paradigm tackles the sample complexity challenge by searching for “large margin” separators. Roughly speaking, a halfspace separates a training set with a large margin if all the examples are not only on the correct side of the separating hyperplane but also far away from it.

Restricting the algorithm to output a large margin separator can yield a small sample complexity even if the dimensionality of the feature space is high (and even infinite) [29]. Finding a hyperplanea line or plane in high-dimensional space—that best distinguishes between distinct classes of data points is the basic goal of SVM. The decision boundary is the name given to this hyperplane. The SVM technique works well in high-dimensional spaces and is also successful when there are more samples than dimensions.

There are two main types of SVM:

Linear SVM: When there is a possibility that the data will be separated by a plane or a straight line, linear SVM is used to solve the classification problem. A linear SVM identifies the hyperplane with the greatest margin of separation between all the different classes of data points.

Non-Linear SVM: In non-linear classification issues where the data cannot be divided by a straight line or a plane, non-linear SVM is utilized. The kernel trick is a method that Non-Linear SVM uses to translate the input data into a higher-dimensional space where it is linearly separable. The polynomial, the radial basis function (RBF), and the sigmoid are a few examples of non-linear kernel functions.

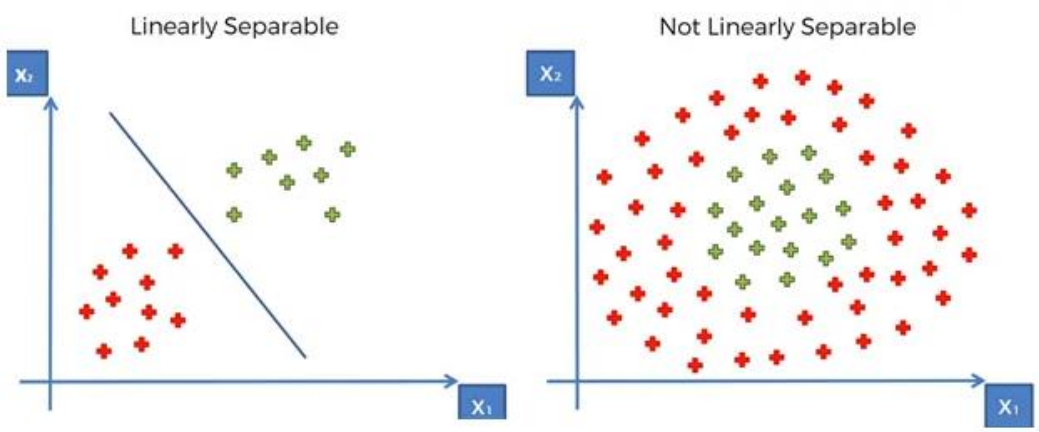

Fig 3. 5 Difference of Linear SVM and Non-Linear SVM [30].

Hyperplane and Support Vectors in the SVM algorithm:

Hyperplane: In n-dimensional space, there may be several lines or decision boundaries used to separate the classes, but we must identify the optimum decision boundary that best aids in classifying the data points. The hyperplane of SVM is a name for this optimal boundary.

The dataset's features determine the hyperplane's dimensions, therefore if there are just two features (as in the example image), the hyperplane will be a straight line. Additionally, if there are three features, the hyperplane will only have two dimensions. It is always build a hyperplane with a maximum margin, or the greatest possible separation between the data points.

Support Vectors: Support vectors are the data points or vectors that are closest to the hyperplane and have the greatest impact on where the hyperplane is located. These vectors are called support vectors because they support the hyperplane.

3.3 Deep Learning and Neural Network Methods

3.3.1 Multilayer Perceptron (MLP)

A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). When referring to networks constructed from many layers of perceptrons, the term "MLP" is sometimes used broadly to refer to any feedforward ANN.

The phrase "multilayer perceptron" does not describe a single perceptron with numerous layers. Instead, it is made up of numerous layers of perceptrons. "Multilayer Perceptron Network" is an option. True perceptrons are a particular type of artificial neuron that has a threshold activation function. MLP perceptrons have a wide range of activation functions available. A MLP neuron has the option to either do classification or regression, based on its activation function, whereas a genuine perceptron performs binary classification [31].

The multilayer perceptron is a model for a nonlinear mapping between an input vector and an output vector that is made up of a network of straightforward linked neurons, or nodes, as shown in Fig. X. An MLP has an input layer, a hidden layer, and an output layer, which together compose at least three layers of nodes. Each node, with the exception of the input nodes, is a neuron that employs a nonlinear activation function. The neurons in a Multilayer Perceptron can employ any arbitrary activation function, in contrast to neurons in a Perceptron, which must have an activation function that enforces a threshold, such as ReLU or sigmoid.

Backpropagation is a supervised learning method that is used by MLP during training. MLP differs from a linear perceptron due to its numerous layers and non-linear activation. It can discriminate between data that cannot be separated linearly. MLPs are useful in research because they can handle stochastic problem-solving.

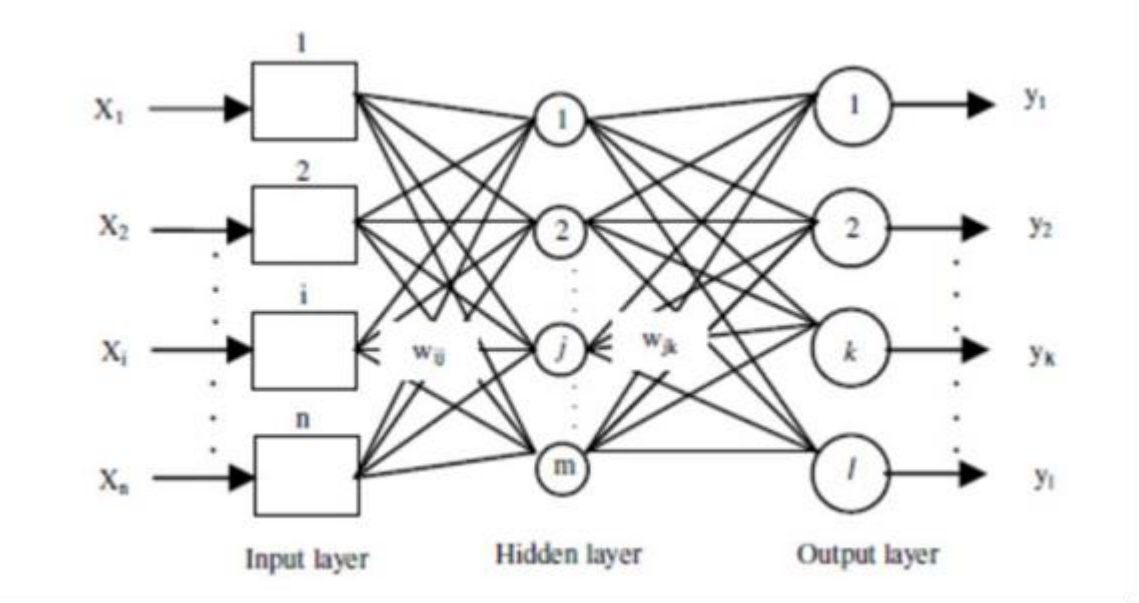

Fig. 3. 6 A Multilayer Perceptron with Two Hidden Layers

The Multilayer Perceptron algorithm falls within the category of feedforward algorithms since inputs are combined with starting weights and added to the activation function, just like in a perceptron, to create a weighted total. The distinction, though, is that every linear combination is carried over to the layer after it. Each layer's calculation output and internal data representation are passed on to the layer below it. This concludes at the output layer after traversing all hidden layers. It has been demonstrated that a multilayer perceptron can approximate any smooth, measurable function between the input and output vectors by choosing an appropriate collection of linking weights and transfer functions.

Multilayer perceptrons can acquire new abilities through practice. A collection of training data is needed, which is made up of input and corresponding output vectors. The training data is repeatedly fed into the multilayer perceptron, and the weights in the network are changed until the desired input—output mapping is achieved. Multilayer perceptrons are supervised in the process of learning. For a specific input vector, the multilayer perceptron's output during training may not match the expected output. The disparity between the desired and actual output is referred to as an error signal. The amount of this error signal is used during training to decide how much the weights in the network should be changed in order to lower the multilayer perceptron's overall error. A multilayer perceptron can be trained using a variety of approaches. The multilayer perceptron can generalize to fresh, unknown input data after being trained with representative training data.

$$a_j = \sum_{i=1}^{d} w_{ji}^{(1)} x_i + w_{j0}^{(1)}. \tag{3.5}$$

Wij(1) denotes a a weight in the first layer, going from input i to hidden unit j, and wj0(1) denotes the bias for hidden unit j.

$$a_j = \sum_{i=0}^{d} w_{ji}^{(1)} x_i. \tag{3.6}$$

The activation of hidden unit j is then obtained by transforming the linear sum in (3.6) using an activation function g(.) to give

$$z_j = g(a_j). \tag{3.7}$$

The outputs of the network are obtained by transforming the activations of the hidden units using a second layer of processing elements. Thus, each output unit k1 it is constructed a linear combination of the outputs of the hidden units of the form

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)}. \tag{3.8}$$

$$a_k = \sum_{j=0}^{M} w_{kj}^{(2)} z_j \tag{3.9}$$

Which can be represented diagrammatically by including an extra hidden unit with activation z0=1 as shown in figure (3.5). The activation of the kth output unit is then obtained by transforming this linear combination using a non-linear avtivation function,

$$y_k = \widetilde{g}(a_k). \tag{3.10}$$

g(.) notation is used for the activation function. (3.7), (3.8), (3.9), (3.10) is combined

$$y_k = \widetilde{g}\left(\sum_{j=0}^{M} w_{kj}^{(2)} g\left(\sum_{i=0}^{d} w_{ji}^{(1)} x_i\right)\right) \tag{3.11}$$

(Hyperbolic tangent sigmoid transfer function (tansig)), weight and bias.

3.3.2 Recurrent Neural Networks (RNN)

Recurrent neural networks or RNNs are a family of neural networks for processing sequential data [33]. Data from training sets are used to train recurrent neural networks. Because they use data from earlier inputs to affect the present input and output, they are distinguished by their "memory". Recurrent neural networks' outputs are dependent on the previous parts in the sequence, unlike typical deep neural networks, which presume that inputs and outputs are independent of one another. Unidirectional recurrent neural networks are unable to take into account future events in their forecasts, despite the fact that they would be useful in deciding the output of a particular sequence.

Recurrent networks are distinguished by the fact that each layer of the network uses the same parameters. Recurrent neural networks share the same weight parameter inside each layer of the network, in contrast to feedforward networks, which have distinct weights across each node. However, to support reinforcement learning, these

weights are still modified using the techniques of backpropagation and gradient descent.

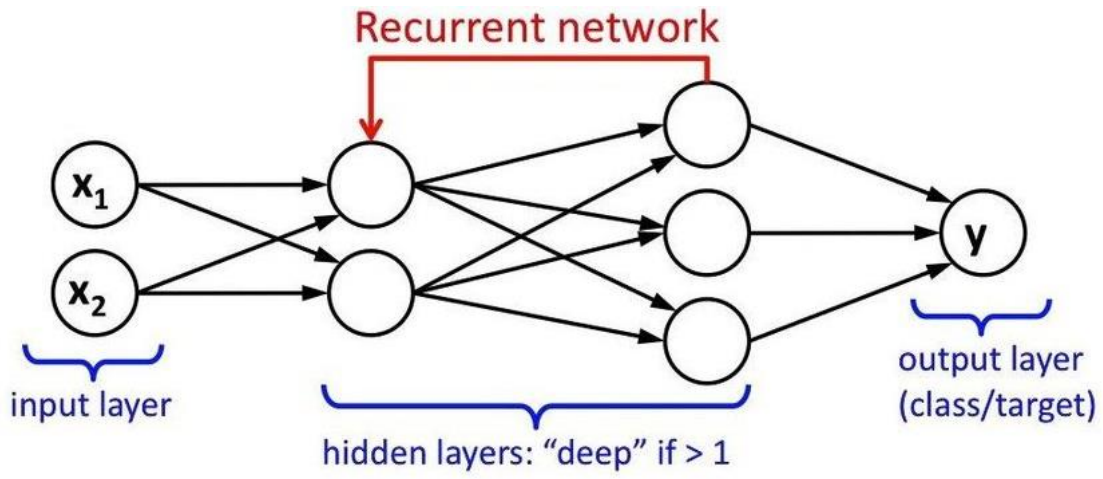

Fig. 3. 7 RNN Structure

Recurrent neural networks use the backpropagation through time (BPTT) algorithm, which differs slightly from conventional backpropagation because it corresponds to sequence data, to find the gradients. In traditional backpropagation, the model trains itself by computing errors from its output layer to its input layer. These computations enables to accurately alter and fit the model's parameters. In contrast to feedforward networks, which do not share parameters between layers, BPTT adds errors at each time step, which is how it varies from the conventional technique.

RNNs frequently experience the two issues of "exploding gradients" and "vanishing gradients" throughout this process. The gradient's size, or the slope of the loss function along the error curve, is what categorizes these problems. The weight parameters are updated until the gradient is too small, at which point they are no longer meaningful. The algorithm stops learning when something happens. When the gradient is too great, exploding gradients happen, which makes the model unstable. In this scenario, the model weights will eventually become too enormous and be represented as NaN. Reducing the number of hidden layers in the neural network and hence some of the complexity in the RNN model is one way to address these problems.

As shown in Fig. 3. 7, an RNN consists of internal states, inputs, outputs, weights, activation functions, and feedback links. The current states are determined by the previous states, weights, and inputs as the following Eqs. (3.12), (3.13), and (3.14).

$$\hat{h}_j = f\left(\sum_{i=1}^{n} v_{ji}\hat{x}_i(q) + \sum_{i=n+1}^{n+m} v_{ji}u_{i-n}(q-d)\right) \tag{3.12}$$

$$\hat{x}_k(q+1) = \sum_{j=1}^{n} w_{kj}\hat{h}_j(q+1) \tag{3.13}$$

$$\hat{y}(q+1) = \sum_{j=1}^{i} w_{n+1j}\hat{h}_j(q+1) \tag{3.14}$$

Where f is sigmoid function, vji and wkj are weights between input layer and hidden layer, and hidden layer and output layer, respectively. The steepest gradient method is used for the learning [12].

3.3.3 Long-Short Term Memory (LSTM)

Long Short-Term Memory Networks is a deep learning, sequential neural network that permits information to endure. It is a specific variety of recurrent neural network that can address the vanishing gradient issue that RNNs encounter. Hochreiter and Schmidhuber [35] created LSTM to address the issue with conventional rnns and machine learning techniques. Python's Keras package can be used to implement LSTM.

These blocks can be compared to differentiable memory chips found in digital computers. Three multiplicative units—the input output and forget gates—provide continuous analogs of write, read, and reset operations for the cells, and each block comprises one or more self-connected memory cells [35].

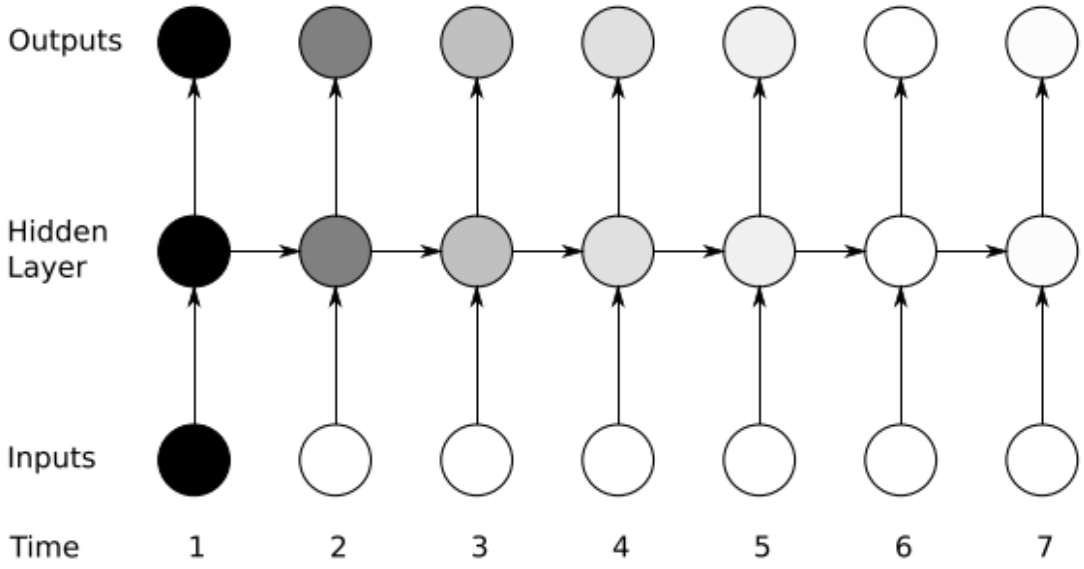

Fig. 3. 8 The Vanishing Gradient Problem for Rnns

LSTM networks are composed of an input layer, one or more hidden layers, and an output layer. The number of neurons in the input layer is equal to the number of explanatory variables (feature space). The number of neurons in the output layer reflects the output space, i.e., two neurons in our case indicating whether or not a stock outperforms the cross-sectional median in t + 1. The main characteristic of LSTM networks is contained in the hidden layer(s) consisting of so called memory cells. Each of the memory cells has three gates maintaining and adjusting its cell state st: a forget gate (ft), an input gate (it), and an output gate (ot). The structure of a memory cell is illustrated in Fig. 3. 8.

The nodes in the unfolded network are shaded, and the darker the shade, the more sensitive they are to the inputs at time one. The sensitivity decreases with time as fresh inputs replace hidden layer activations and the network "forgets" the initial inputs.

The weights of the input and recurrent connections are represented in the equations below by the matrices Wq and Uq, respectively. Depending on the estimated activation, the subscript q may represent the input gate i, output gate o, forget gate f, or memory cell c. Note that vectors are represented by lowercase variables. Ct ∈ R h contains the units of h LSTM cells rather than just one unit from one LSTM cell. The following equations have compact versions for an LSTM cell's forward pass when equipped with a forget gate [36].

$$f_t = \sigma_g(W_f x_t + U_f H_{t-1} + b_f) \tag{3.15}$$

$$i_t = \sigma_g(W_i x_t + U_i H_{t-1} + b_i) \tag{3.16}$$

$$o_t = \sigma_g(W_o x_t + U_o H_{t-1} + b_o) \tag{3.17}$$

$$\tilde{c}_t = \sigma_c(W_c x_t + U_c H_{t-1} + b_c) \tag{3.18}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \tag{3.19}$$

$$h_t = o_t \cdot c_h(c_t) \tag{3.20}$$

Where c0 = 0 and h0 = 0 are the beginning values and operator stands for the Hadamard product (also known as the element-wise product). The time step is indexed by the subscript t.

3.3.4 Echo State Networks (ESN)

A typical ESN has a large sparsely connected reservoir, in which the internal connected weights remain unchanged. Only the connections from the reservoir to the output layer are trainable by linear regression algorithms. Due to the above-mentioned advantages, ESNs attract increasing attention in supervised learning [37].

Echo state networks (ESNs) are first proposed by Jaeger and Hass [38] to learn nonlinear systems and predict chaotic time series. A larger-than-normal layer of neurons, called dynamic reservoir, is used in ESNs with random initialization. All the weight matrices to the reservoir (internal recurrent connections Wx and input-to-hidden layer connections Win) are randomly generated and not adaptable, whereas the reservoir-to-output weight matrix Wout is adapted by supervised learning [37].

Comparing with traditional neural networks, ESN has the following advantages. First, the state of ESN contains information about the past input history in a way which reflects the recent history well and decays with the delay time. This feature of short-term memory solves the problem to add memory and determine the range of time delay values that many previous neural networks meet. Second, the training of ESN is

very simple and it can get the global optimal parameters. Therefore, ESN does not need to worry about local convergence that conventional neural networks often confront with. Third, ESN performs much better than past neural network in stochastic time series prediction [40].

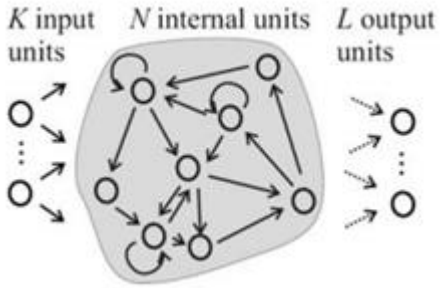

Fig. 3. 9 The Architecture of ESN [40]

ESNs' dynamic behavior is described by the following equations:

$$\mathbf{x}(n) = \boldsymbol{\phi}(\mathbf{W}_x\mathbf{x}(n-1) + \mathbf{W}_{\text{in}}\mathbf{u}(n)) \tag{3.21}$$

$$\mathbf{y}(n) = \mathbf{W}_{\text{out}}^{T}\mathbf{x}(n) \tag{3.22}$$

Where x(n) is a q × 1 vector, u(n) and y(n) are m × 1 vectors, and y(n) = u(n + 1). Wx is a q × q matrix, Win and Wout are q × m matrices, and ϕ: Rq → Rq is described by

$$\boldsymbol{\phi} : [x_1, x_2, \ldots, x_q]^T \rightarrow [\varphi(x_1), \varphi(x_2), \ldots, \varphi(x_q)]^T. \tag{3.23}$$

The nonlinear function ϕ(·) takes the form

$$\varphi(x) = \tanh(x) = (1 - e^{-2x})/(1 + e^{-2x}). \tag{3.24}$$

For some tasks, there are recurrent connections between the reservoir and the readout neurons, which are realized by feedback connection Wfb from the trained output y(n − 1) to the reservoir state x(n). Note that for time series prediction, u(n) is equal to y(n − 1), so u(n) and Win are the equivalent of y(n − 1) and Wfb, which are just a matter of notation [37].

The input weights Win are randomly initialized in the range [−γ,γ], and γ is called the input scaling parameter. Internal recurrent connections Wx are sparsely connected, and its spectral radius ρ(Wx) (the absolute value of the largest eigenvalue of Wx) is usually less than 1 to ensure the echo state property, but this is neither a sufficient condition nor a necessary condition. The spectral radius could be considered as a task-dependent global parameter for optimization [37].

Least square estimation can be applied to compute Wout, since no cyclic dependence exists in the trained readout connections. It is dismissed that the states x(n) from the beginning of the training process, since they are affected by the initial conditions. To make the notation simple, it assumed that they come before n = 1. Collect the remaining echo states xT (n) row wise into a matrix X

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^T(1) \\ \mathbf{x}^T(2) \\ \vdots \\ \mathbf{x}^T(p) \end{bmatrix} = \begin{bmatrix} x_1(1) & x_2(1) & \cdots & x_q(1) \\ x_1(2) & x_2(2) & \cdots & x_q(2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(p) & x_2(p) & \cdots & x_q(p) \end{bmatrix}. \tag{3.25}$$

Denote Xi as [xi(1) xi(2) ⋯ xi(p)] T , i = 1, 2,..., q. Xi is the state of the ith node, and X could be rewritten as,

$$\mathbf{X} = [\mathbf{X}_1 \quad \mathbf{X}_2 \quad \cdots \quad \mathbf{X}_q]. \tag{3.26}$$

Make the corresponding target output value yT (n) row wise into a matrix Y = [y(1),y(2)⋯ y(p)]T. The least square estimation of Wout is,

$$\hat{\mathbf{W}}_{\text{out}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}. \tag{3.27}$$

3.4 Performance Measurements and Metrics

3.4.1 Mean Square Error (MSE)

The average of the squares of the errors, or the average squared difference between the estimated values and the actual value, is measured by the mean squared error (MSE) or mean squared deviation (MSD) of an estimator in statistics. MSE, which corresponds to the expected value of the squared error loss, is a risk function. Because of randomness or because the estimator failed to consider information that could have led to a more accurate estimate, MSE is nearly always strictly positive. Empirical risk minimization, or MSE, is a term used in machine learning to describe the empirical risk as an estimate of the real MSE. A measure of an estimator's quality is the MSE. It is always a positive number that gets smaller as the error gets closer to zero since it is derived from the square of the Euclidean distance.

It calculates the average squared difference between the expected and actual values, giving larger differences greater importance. In contrast to other error metrics like Mean Absolute Error (MAE), MSE is hence more susceptible to outliers. The result of squaring the discrepancies is that greater errors are penalized more severely.

The formula for calculating MSE is as follows:

$$MSE = (1/n) * \Sigma(y_i - \hat{y}_i)^2 \tag{3.28}$$

where:

n: is the number of data points

y_i: represents the actual (observed) value of the dependent variable for the i-th data point

$\hat{y}_i$: represents the predicted value of the dependent variable for the i-th data point

Σ: represents the summation symbol, indicating that the squared differences are summed over all data points.

MSE is frequently used in a variety of areas, particularly in regression assignments, to evaluate and compare the effectiveness of alternative models or methods. It offers a numerical assessment of how well a model fits the data, enabling evaluation of model correctness and comparison of competing models.

One of the most popular loss functions in statistics is squared error loss, although its popularity is largely due to mathematical ease than actual loss considerations in practical applications. MSE is a useful mathematical tool for examining the performance of linear regression since it allows one to separate the variation in a dataset into that explained by the model and that explained by chance.

3.4.2 Root Mean Square Error (RMSE)

The Mean Squared Error (MSE), which is a common evaluation metric for regression issues, is the source of RMSE. The square root of the average of the squared differences between the estimated amounts and the actual values is used to calculate RMSE.

The formula for calculating RMSE is as follows:

$$RMSE = \sqrt{((1/n) * \Sigma(y_i - \hat{y}_i)^2)} \quad (3.29)$$

where:

n: is the number of data points

y_i: represents the actual (observed) value of the dependent variable for the i-th data point

$\hat{y}_i$: represents the predicted value of the dependent variable for the i-th data point

Σ: represents the summation symbol, indicating that the squared differences are summed over all data points.

A measurement of the average size of the prediction errors in the same units as the dependent variable is provided by RMSE. By calculating the MSE's square root, RMSE restores the error metric to the data's original scale, making it simpler to understand and connect to actual values.

When errors have a normal distribution and when larger errors have a greater impact, RMSE is especially helpful. It is a frequently used metric for evaluating the effectiveness of regression models and contrasting various algorithms or models in several disciplines, including statistics, machine learning, and data analysis.

Similar to MSE, RMSE values under 0 indicate a perfect fit to the data. Lower values of RMSE indicate greater model performance. You can identify which model or algorithm offers a more accurate prediction of the dependent variable by comparing the RMSE of various models or algorithms.

3.4.3 R-Squared (R2)

The coefficient of determination, or R-squared, is a statistical indicator that shows how much of the variance in the dependent variable can be accounted for by the independent variables in a regression model. It is a widely used metric to evaluate a regression model's goodness-of-fit. The correlation coefficient (r) between the expected values and the actual values of the dependent variable can also be used to calculate R-squared.

The formula for calculating R-squared is as follows:

$$R^2 = ESS / TSS \tag{3.30}$$

where:

ESS: is the explained sum of squares, which measures the total variation in the dependent variable that is explained by the regression model.

TSS: is the total sum of squares, which measures the total variation in the dependent variable.

Higher numbers indicate a better fit of the model to the data, and R-squared proceeds from 0 to 1. A number of R-squared of 0 implies that the independent variables explain no variance in the dependent variable, whereas a value of R-squared of 1 indicates that they explain for all variance.

It's crucial to keep in mind that R-squared has some restrictions. It gives us no information regarding the coefficients' statistical significance or the accuracy of the forecasts. By including extra independent variables in the model, even if they are meaningless or irrelevant, R-squared can be unnaturally inflated. Therefore, when analyzing a regression model, it is crucial to take into account additional metrics and diagnostic tools in addition to R-squared.

3.4.4 Mean Absolute Error (MAE)

Mean Absolute Error is referred to as MAE. It is a typical indicator used to evaluate how successfully regression models perform. The average absolute difference between the expected and actual values is measured by MAE.

The formula for calculating MAE is as follows:

$$MAE = (1/n) * \Sigma|y_i - \hat{y}_i| \tag{3.31}$$

where:

n: is the number of data points.

y_i: represents the actual (observed) value of the dependent variable for the i-th data point

$\hat{y}_i$: represents the predicted value of the dependent variable for the i-th data point

Σ: represents the summation symbol, indicating that the absolute differences are summed over all data points.

The average size of the prediction errors, measured by MAE, is given in the same units as the dependent variable. Since MAE does not square the variations between predicted and actual values, it fails to penalize large errors more severely than MSE (Mean Squared Error) or RMSE (Root Mean Squared Error).

Because it takes into account the absolute differences, MAE is robust against outliers, making it a useful indicator when outliers are present. Since MAE measures the average absolute deviation of the predictions from the actual values, it is also simple to understand.

A MAE of 0 indicates a perfect fit to the data, while lower values of MAE, like other evaluation measures, imply greater model performance. MAE is frequently used to evaluate the precision of regression models and contrast alternative techniques or models in a variety of areas, including machine learning, economics, and finance.

CHAPTER 4

APPLICATION OF INVENTORY AGE PREDICTION

4.1 Data Collection

Real life dataset of 3 warehouses for an alcoholic beverage company is used for application of models for inventory age prediction.

4.1.1 Data Features and Contents

The data consists of 14 features such as warehouse number, goods, group of goods, definitions of goods and group of goods, quantity, unit, net liter, delivery number, quantity liter, shelf life, storage time, shelf life percentage and inventory age as it can be seen in Tab 4. 1.

Tab 4. 1 Data and Features

DNo	MG	MG Definition	Goods	Goods Definition	Quantity	Unit	Net Liter	Quantity Lt	Delivery No	Shelf Life	Storage Time	Shelf Life Perc	Inventory Age
Izmir					144.00	KAS	10.00	1440.00	322890	180.00	135.00	1.66	3.00
Adana					117.00	KAS	10.00	1170.00	322890	180.00	135.00	1.66	3.00
Izmir					495.00	KAS	10.00	4950.00	322890	180.00	135.00	1.66	3.00
Ankara					396.00	KL	10.00	3960.00	323261	120.00	75.00	8.94	11.00
Izmir					132.00	KL	10.00	1320.00	323261	180.00	135.00	17.58	32.00
Ankara					198.00	KL	10.00	1980.00	323261	180.00	135.00	8.24	15.00
Izmir					66.00	KL	10.00	660.00	323261	180.00	135.00	7.14	13.00
Ankara					132.00	KL	10.00	1320.00	323261	180.00	135.00	7.14	13.00
Izmir					20.00	KL	7.92	158.40	322834	270.00	210.00	21.98	60.00
Izmir					10.00	TVA	12.00	120.00	322834	270.00	210.00	17.95	49.00
Adana					10.00	TVA	12.00	120.00	322834	120.00	75.00	23.58	29.00
Izmir					90.00	TVA	12.00	1080.00	322834	180.00	135.00	13.74	25.00
Izmir					90.00	TVA	12.00	1080.00	322834	180.00	135.00	22.10	40.00
Ankara					27.00	KL	7.20	194.40	322834	180.00	135.00	2.76	5.00
Izmir					20.00	PAK	7.20	144.00	322834	180.00	135.00	14.84	27.00
Izmir					30.00	KL	8.00	240.00	322834	180.00	135.00	17.03	31.00
Adana					90.00	TVA	12.00	1080.00	322834	180.00	135.00	15.38	28.00

Warehouse Number: "DNo" is called as warehouse number. There are 3 warehouses which are planted in Izmır, Ankara and Adana based on the critical locations. To be able to convert this data to numeric numbers each city is assigned to number codes as 1, 2 and 3 in the same order.

Goods and definitions: There are 62 different goods and each one has unique number code and definitions to easily separate one from each other. Even though some of these goods are same brand, they are distinct by types of batches. These types consist of names like returnable, non-returnable, multi-pack, can and bottle.

Group of goods and definitions: Each good is assigned to a group based on brand, type and unit of goods. There 19 different groups for selection.

Quantity: There is a wide range of batches that are sold for these warehouses. Therefore, quantity of goods varies between 1 and 1980. The quantity of can change depending on features like unit type, net liter of the good and delivered location.

Unit: The alchologic beverages consists of 4 types in this data. These types are differ for certain products type, it is decided by company's strategic and planning departments.

Net Liter: Amount of liter in a single quantity good.

Quantity Liter: Quantity liter is the total amount of liter in a batch of goods. It is calculated by multiplying quantity and net liter features.

Delivery Number: Delivery number is a unique value for each bach of good.

Shelf Life: Storage time is the total amount of time for spesific type of good can be stored before the expiration date.

Storage Time: Only difference between this topic and shelf life is, storage time is the maximum amount of time for spesific type of good can be stored based on the company and health regulations.

Inventory Age: Recorded inventory age of each variable data.

Tab 4. 2 Data Feature Statistics

	Quantity	Net	Liter	Shelf Life	Storage Time	Inventory Age	MG	Goods	Unit	Delivery No
count	4916.00	4916.00	4916.00	4916.00	4916.00	4916.00	4916.00	4916.00	4916.00	4916.00
mean	168.90	9.98	1751.87	189.81	142.48	15.95	5.34	22.76	1.71	258.60
std	242.33	2.25	2705.60	37.67	32.76	12.73	4.88	15.72	1.23	167.37
min	1.00	5.69	6.00	120.00	75.00	1.00	0.00	0.00	0.00	0.00
25%	30.00	8.00	285.12	180.00	135.00	6.00	0.00	12.00	1.00	112.75
50%	90.00	10.00	864.00	180.00	135.00	12.00	6.00	20.00	1.00	243.00
75%	180.00	12.00	2160.00	180.00	135.00	23.00	8.00	35.00	3.00	400.00
max	1980.00	12.00	23760.00	270.00	210.00	59.00	17.00	56.00	3.00	570.00

4.1.2 Data Correlation

After understanding the content of the data found and selecting the target label to be predicted, some features were manually eliminated by comparing at the relationships between the features and the target label. These features are: MG Definition and Goods Definition. It has been seen that these features should be removed since they are the same features as "MG" and "Goods column", and this will not affect the success of the estimation.

After removed "Goods Definition" and "MG Definition" columnes, correlation matrix is plotted to investigate relationships between target label that will be predicted and

features. According to statistics, correlation values closer to +1 indicate a strong positive correlation, values closer to -1 indicate a strong negative correlation, and values near 0 indicate a weak or no correlation. In the light of such information, features that is “DNo” and with a correlation value close to 0 with inventory age is removed. When the correlation matrix was examined, it was seen that there were correlation values close to +/- 0.2 on average between the target label and the features. For this reason, each of the features with a correlation value of 0.1 was removed one by one and their effects on the prediction success were observed. After this process, features with correlation value of +/- 0.1 were decided to remain in the data matrix to get better estimation.

Also, the correlation matrix shows that the relationship between inventory age and shelf life percentage is a strong positive correlation. This correlation value can cause overfit problems during predicting test values so, this feature was removed.

Besides, It was noticed that the correlation value between shelf life and storage time was 1. This value indicates a strong positive correlation. Despite this correlation value, as explained in the data description section, although these two features seem to be the same, it was decided to keep these features in the data matrix because they describe different time periods. These features also increase the prediction success.

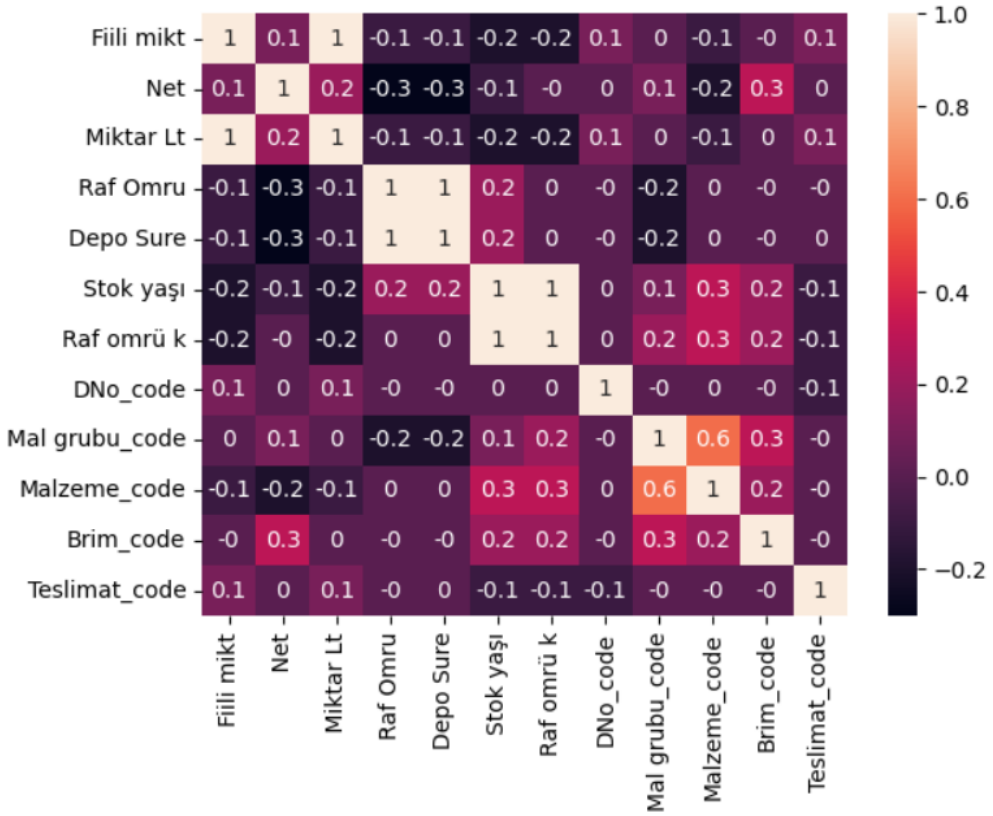

Fig 4. 1 Correlation Matrix

After the feature evaluation and selection part was done, 10 features remained in the data matrix with the target value, the inventory age.

DNo_Code	MG	Goods	Quantity	Unit_Code	Net_Liter	Quantity_Lt	Delivery_No	Shelf_Life	Storage_Time	Inventory_Age
1			144.00	1	10.00	1440.00	322890	180.00	135.00	3.00
1			117.00	1	10.00	1170.00	322890	180.00	135.00	3.00
2			495.00	1	10.00	4950.00	322890	180.00	135.00	3.00
3			396.00	2	10.00	3960.00	323261	120.00	75.00	11.00
1			132.00	2	10.00	1320.00	323261	180.00	135.00	32.00
1			198.00	2	10.00	1980.00	323261	180.00	135.00	15.00
1			66.00	2	10.00	660.00	323261	180.00	135.00	13.00
1			132.00	2	10.00	1320.00	323261	180.00	135.00	13.00
2			10.00	3	12.00	120.00	322834	270.00	210.00	49.00
1			10.00	3	12.00	120.00	322834	120.00	75.00	29.00
3			90.00	3	12.00	1080.00	322834	180.00	135.00	25.00
2			90.00	3	12.00	1080.00	322834	180.00	135.00	40.00
2			27.00	2	7.20	194.40	322834	180.00	135.00	5.00
1			20.00	4	7.20	144.00	322834	180.00	135.00	27.00
2			30.00	2	8.00	240.00	322834	180.00	135.00	31.00
1			90.00	1	12.00	1080.00	322834	180.00	135.00	28.00
3			90.00	1	12.00	1080.00	322834	180.00	135.00	29.00
3			20.00	1	7.92	158.40	322834	180.00	135.00	52.00
3			20.00	1	7.92	158.40	322834	180.00	135.00	52.00
1			20.00	1	7.92	158.40	322834	180.00	135.00	51.00

Fig 4. 2 Final State of The Data Matrix

4.1.3 Data Visualization

First of all, in the data visualization part, the distribution of the target value, the inventory age, was examined. Thus, it was observed whether it was statistically suitable for a certain distribution. According to this table, although the data are not suitable for a certain distribution, it is seen that the inventory age data is in the majority in the range of 0 to 35.

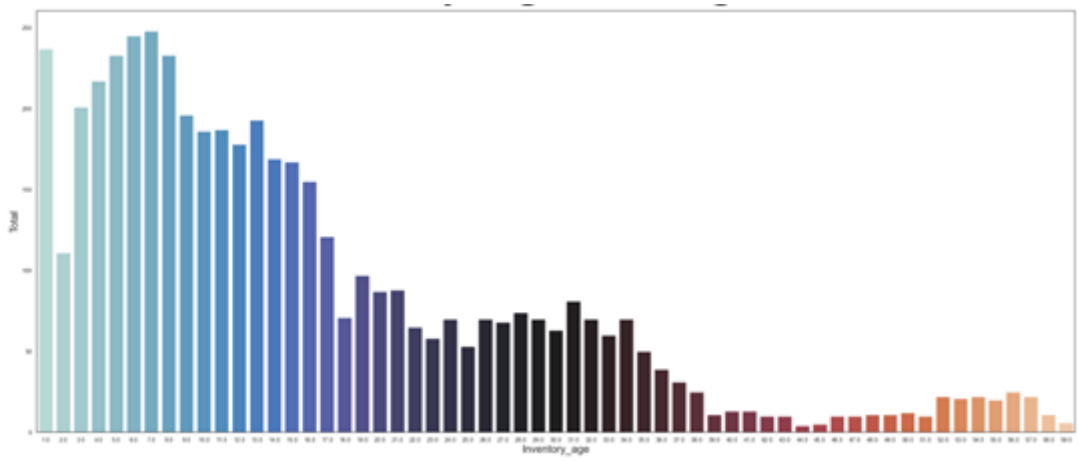

Fig 4. 3 Inventory Age Histogram

Secondly, since there are 4 types in the unit feature, a table was made for this feature, and its distribution and estimation were investigated. As a result of the examination, it was observed that the 2nd type unit may cause deviations in the estimation.

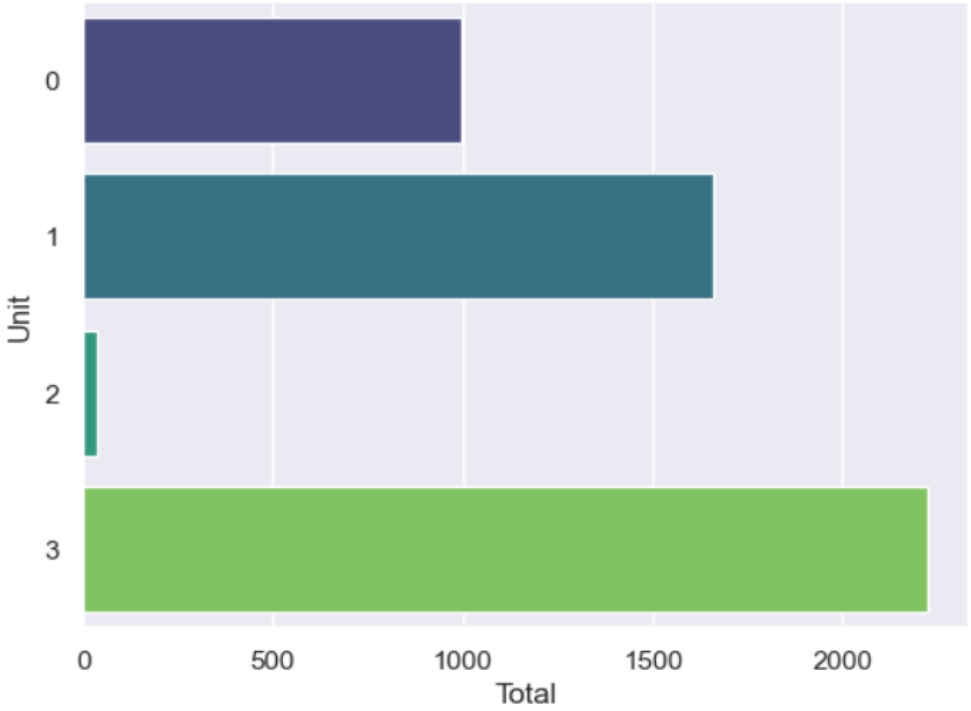

Fig 4. 4 Unit Histogram

Thirdly, since there are 3 options as 75, 135 and 210 in the storage time feature, a table has been made for this feature and the distribution and estimation of this data has been examined. As a result of the examination, it was observed that the density of the 135 value in the data was too high and it was seen that this could cause deviations in the estimation.

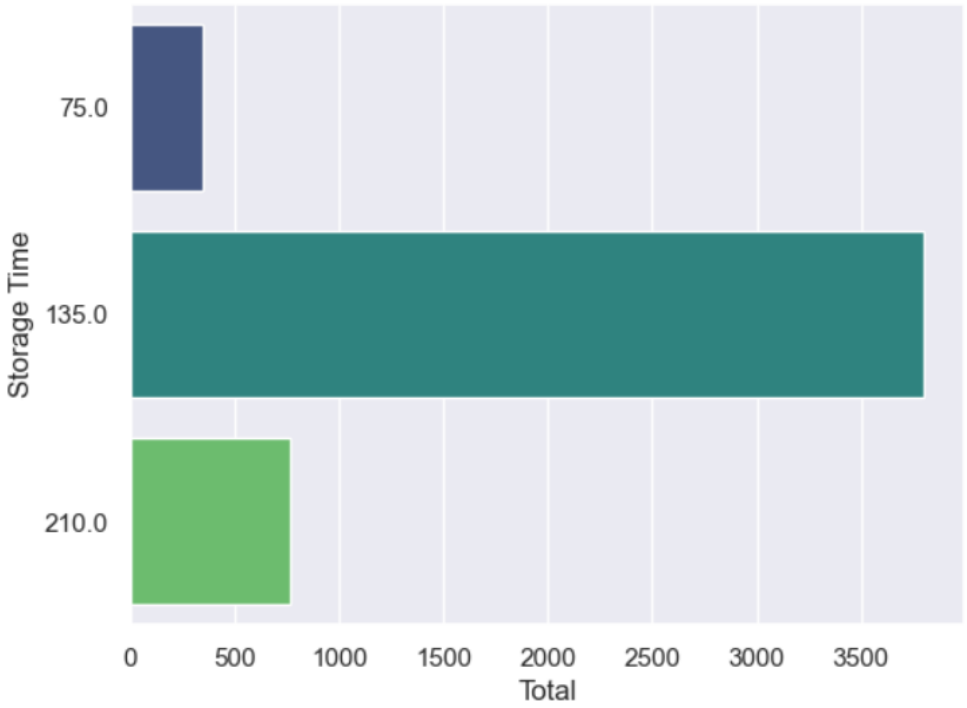

Fig 4. 5 Storage Time Histogram

Finally, the density graph was used due to the wide range of values in the liter feature. From this graph, it was seen that the liter data were collected in a smaller range. It is understood that this will have a high effect on the prediction model.

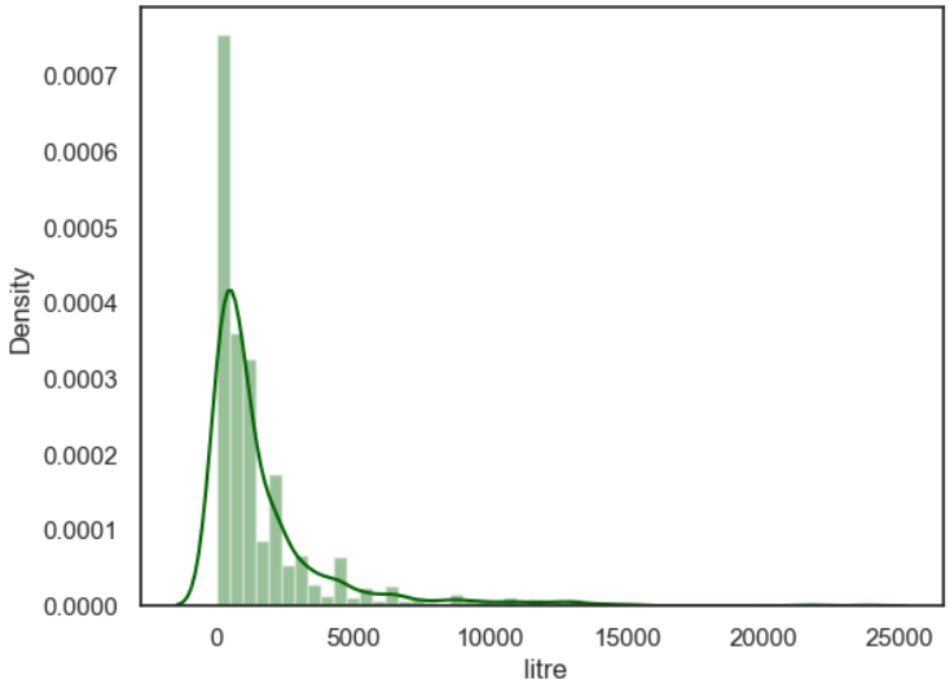

Fig 4. 6 Liter Density Graph

4.2 Application of Models with Python

4.2.1 Machine Learning Applications

Simple Linear Regression:

A number of steps are taken in order to implement a Simple Linear Regression algorithm in Python. First, the essential libraries are loaded, including Scikit-Learn for machine learning capabilities, Pandas for data processing, and NumPy for numerical calculations. After that, a pandas command is used to load the dataset. After that as it was explained in feature selection part, data is cleaned and some unnecessary feature are dropped. For the purpose of reading and calculating data, LabelEncoder() converts string data to float. To prepare for training and evaluation, the data are divided into independent variables (X) and the dependent variable (y). Standardization and the elimination of scale effects are accomplished with StandartScaler(). To make model training and performance evaluation easier, the data is then split into a training set and a testing set. 2/3 of the data are utilized for training the dataset, while 1/3 are used for testing. The 'LinearRegression' class from scikit-learn must be implemented in order to generate a linear regression model. By calling the 'fit()' method on the Linear Regression object and providing the inputs x_train and y_train, the model is trained using the training data. Metrics like MSE, RMSE, MAE, and R-squared score are taken into consideration while assessing the model's performance using the testing set. Finally, predictions are made on new, unseen data by using the trained model's "predict()" method while passing x_test. Considering the R2 score which is critically poor with 0.22, it was observed that the linear regression algorithm is not sufficient for the problem and it is necessary to solve it with a more complex model.

```
MSE: 121.48882521197753
RMSE: 11.02219693218995
R-squared: 0.23314207664828324
MAE: 8.672524210902202
```

Fig 4. 7 Test Scores of Linear Regression

Polynomial Regression with Ridge Regression:

As mentioned earlier, the essential libraries are imported, dataset is loaded, data is divided into a training set and testing set and feature selection is applied. The 'PolynomialFeatures' class from scikit-learn is used to convert the input features into polynomial features during feature engineering. By instantiating the 'Ridge' class and providing the desired regularization parameter (alpha), a polynomial regression model with ridge regularization can be constructed. The modified polynomial features are used as inputs (x_train) together with the corresponding target value (y_train) and the model is then trained using the training data by using the 'fit()' function on the Polynomial Regression object. Using the testing set, the model's performance is evaluated using metrics like MSE, RMSE, MAE, and R-squared score. Finally, predictions on new, unseen data may be made by using the trained model's "predict()" method and passing it the new data's modified polynomial features (x_test). Considering the

R2 score which is better results than linear regression with 0.59, it was observed that the polynomial regression with ridge regression model can be used for the problem but and another algorithms may be better to solve.

```
(poly deg 3) R-squared score (training): 0.627
(poly deg 3) R-squared score (test): 0.558

(poly deg 3 + ridge) R-squared score (training): 0.609
(poly deg 3 + ridge) R-squared score (test): 0.594
```

Fig 4. 8 Test Scores of Polynomial Regression and

Poly. with Ridge Regression

Support Vector Machine (SVM):

To implement Support Vector Machine (SVM) in Python, as mentioned earlier, the essential libraries are imported, dataset is loaded, data is divided into a training set and testing set and feature selection is applied. By creating the 'SVC' class from scikit-learn and specifying the needed kernel (linear), an SVM model can be created. When the 'fit()' method of the SVM object is called with the inputs x_train and y_train, the model is then trained using the training data. Finally, predictions on new, unseen data may be made by using the trained model's "predict()" method and passing it the new data's modified SVM features (x_test).Using the testing set, the model's performance is evaluated using metrics like MSE, RMSE, MAE, and R-squared score. Finally, considering the R2 score which is worse results than even linear regression with 0.1, so, other algorithms should be better to solve.

```
R-Squared Score of Linear SVC classifier on training set: 0.12
R-Squared Score of Linear SVC classifier on test set: 0.11
```

Fig 4. 9 Test Scores of SVM

4.2.2 Deep Learning and Neural Networks Applications

Multilayer Perceptron (MLP):

The Multi-Layer Perceptron (MLP) method is implemented in Python by following a set of steps. First, the relevant libraries are loaded, giving access to tools for creating and honing MLP models, such as scikit-learn and Keras. The dataset is then imported and preprocessed as necessary to make sure the data is in the right format with features and labels. The MLP model is then trained using the training set, and its performance is assessed using the testing set. Then data is scaled using a StandardScaler() and then divided into two groups: training set and testing set. The architecture of the model, including the number of layers, alpha number, and the solver type, has been tried for numerous variations. After trying these variations, it is decided that best possible

combination for the model has 10 hidden layers, alpha is set for 0.5 and 'lbfgs' solver is used. To make the model development process easier, scikit-learn's "MLPClassifier" library is used. By adjusting the model's weights and biases using an optimization procedure, such as backpropagation, the MLP model is trained using the training data. The testing set is used to evaluate the model's performance once it has been trained, and metrics like accuracy and precision are computed. Finally, by running current, unobserved data through the trained MLP model, necessary functions are called to calculate prediction scores and display actual and predicted values that can be seen in Fig 4. 12.

```
MSE: 94.99753542821935
RMSE: 9.74666791412426
R-squared: 0.40035955887378827
MAE: 5.476894639556377
```

Fig 4. 10 Test Scores of Multilayer Perceptron

Recurrent Neural Networks (RNN):

There are several stages that is taken in order to develop an RNN (Recurrent Neural Network) algorithm in Python. First, the required libraries, including TensorFlow and Keras are imported. Then, the dataset is loaded and any necessary preprocessing is done to make sure the data is in the right format. Data is scaled using a StandardScaler(). In order to assist model training and assessment, the data is then divided into a training set and a testing set. Next, RNN model is created using Keras' Sequential API. This model is constructed by adding multiple layers to the sequential stack. It starts with a SimpleRNN layer that has 45 units and is set to return sequences rather than a single output. The input shape of this layer is determined by the shape of the training data. Dropout layers with a rate of 0.2 are added after each SimpleRNN layer to regularize the model and mitigate overfitting. This pattern is repeated three more times with additional SimpleRNN and Dropout layers. Finally, a Dense layer with a single unit is added to produce the desired output. The model is then compiled using the Adam optimizer and the mean squared error loss function. The Adam optimizer is an optimization algorithm that adapts the learning rate during training. With the model compiled, the next step involves training the model using the fit function. The training data (x_train and y_train) are provided, and the model is trained for 100 epochs with a batch size of 32. Once the training is complete, the model is utilized to make predictions on the test data (x_test). The predicted values are stored in the variable y_pred. To evaluate the performance of the model, several metrics are calculated. Finally, the results of the evaluation metrics are printed to the console, providing insights into the model's performance.

```
MSE: 51.45666085734378
RMSE: 7.173329830514123
R-squared: 0.675196891410998
MAE: 4.731962193916847
```

Fig 4. 11 Test Scores of RNN

Long-Short Term Memory (LSTM):

To implement an LSTM (Long Short-Term Memory) algorithm in Python, necessary libraries are imported such as NumPy, Pandas, scikit-learn, TensorFlow, Keras, and matplotlib. Then, the dataset is loaded and any necessary preprocessing is done to make sure the data is in the right format. Data is scaled using a StandardScaler(). It then defines a sequence length (SEQ_LEN) with a value of 100. Next, a function named to_sequences is defined, which converts the raw data into sequences of length seq_len. It iterates over the data and creates subsequences of the specified length. The function returns the resulting sequences as a NumPy array. Another function called preprocess is defined to preprocess the raw data by utilizing the to_sequences function. It splits the data into training and testing sets based on a specified train split ratio. The function returns the training and testing sets as NumPy arrays. Data is preprocessed by invoking the preprocess function and providing the scaled data, sequence length (SEQ_LEN), and a train split ratio of 0.80. The resulting training and testing sets are assigned to X_train, y_train, X_test, and y_test, respectively. Subsequently, various variables such as window_len, zero_base, dropout, and output_size are initialized to specific values. The model is then defined using the Sequential API from Keras. It consists of several layers, including Convolution1D, MaxPooling1D, LSTM, and Dense layers. The model is compiled with the mean squared error (MSE) loss function and the Adam optimizer. Then, model is trained using the fit function, where the training data (X_train and y_train), along with the number of epochs, batch size, and validation split ratio, are provided. Once the training is complete, the model is used to predict the output for the testing data (X_test), and the predictions are stored in the variable predicteddata. To evaluate the performance of the model, several metrics are calculated. Finally, the results of the evaluation metrics are printed to the console, providing insights into the model's performance.

```
MSE: 0.08343485223605923
RMSE: 0.2888509169728551
R-squared: -0.6252688921901963
MAE: 0.22055537821832946
```

Fig 4. 12 Test Scores of LSTM

Echo State Network (ESN):

The first step is to import the necessary libraries, including `pyESN`, which provides tools for building and training ESN models. Then, a function is created for demonstrading actual values and predicted values side by side for comparison. The dataset is then loaded and preprocessed as usual. Next, the data is split into a training set and a testing set to facilitate model training and evaluation. An ESN model is created by specifying the desired parameters, such as the number of reservoir neurons, the spectral radius, and the input scaling factor. To instantiate the ESN

model, an `ESN` object is created from the `pyESN` library, using the chosen parameters. The ESN model is trained by calling the `fit()` method on the instantiated `ESN` object, passing the training data as input. The performance of the trained model is evaluated using the testing set, considering metrics such as accuracy or root mean squared error. Finally, predictions can be made on new, unseen data by calling the `predict()` method on the trained `ESN` object. Lastly, necessary functions are called to calculate prediction scores and display actual and predicted values that can be seen in Fig 4. 15.

```
MSE: 25.8968607868563
RMSE: 5.088895831794585
R-squared: 0.8485891553374683
MAE: 0.9486581426985103
```

Fig 4. 13 Test Scores of ESN

	Actual		Predicted
	0		0
0	8.0	0	7.990764
1	24.0	1	24.131321
2	9.0	2	8.961814
3	26.0	3	25.991750
4	20.0	4	18.994441
5	52.0	5	52.024000
6	29.0	6	31.001704
7	6.0	7	5.996421
8	29.0	8	28.987104
9	7.0	9	6.999748

Fig 4.14 Actual and Predicted Values of ESN Model

4.3 Results and Discussion

In this thesis, 7 models are created in Python to calculate inventory age prediction scores on warehouses for a alcoholic beverage company which manages its operations on Eastern Europe and Turkey. The created models are consists of Linear Regression, Polynomial Regression with Ridge Regression, Support Vector Machine (SVM), Multilayer Perceptron (MLP), Recurrent Neural Networks (RNN), Long-Short Term Memory (LSTM), Echo State Networks (ESN). The purpose of this thesis is to minimize the storage cost, which has a significant share in the total cost, by predicting the storage duration of the products in the warehouse. Necessary data were collected and evaluations were made to provide a clearer understanding of the problem.

Tab 4. 3 Final Scores of Each Model

	MSE	RMSE	R-Squared	MAE
Linear Regression	121.49	11.02	0.23	8.67
Polynomial Regression with Ridge Regression	52.13	7.22	0.59	5.25
Support Vector Machine (SVM)	176.25	13.27	0.11	8.86
Multilayer Perceptron (MLP	94.99	9.74	0.4	5.47
Recurrent Neural Networks (RNN)	51.45	7.17	0.68	4.73
Long-Short Term Memory (LSTM)	0.05	0.23	-0.04	0.17
Echo State Networks (ESN)	**25.54**	**5.05**	**0.85**	**0.94**

Firstly, machine learning algorithms are compared by R-squared scores. According to Table 4. 1, polynomial regression with ridge regression has the best score, 0.59, out of all machine learning models. When it comes to the other metrics, polynomial regression with ridge regression has a drastically better results. Therefore, this model is selected out of all the machine learning models. Then, deep learning models are created for requirement of more complex algorithms to achieve better prediction scores. Before comparing deep learning methods, Long-Short Term Memory (LSTM) scores are inspected due to unpredictible scores. LSTM model scores shows that, time series prediction is not suitable for our problem. Altough RMSE and MSE scores are close to zero, which is a favorible result, R-squared score has an imprecise value due to being negative. Because, R-squared has to proceed from 0 to 1. After observing the MSE, RMSE, R-squared and MAE scores, it is seen that Long-Short Term (LSTM) model has a overfitting problem even after adding the cross-validation process. From this perspective, LSTM scores are disregarded from this thesis. After comparising all score metrics, Echo State Network (ESN) has the best prediction scores with, 0.85 r-squared score, among all the deep learning models. For this inventory age problem it has seen that, Echo State Networks (ESN) is the most sufficient algorithm to predict inventory age for alcoholic beverage inventory dataset.

CHAPTER 5

CONCLUSION

In this thesis, machine learning and deep learning models are created using Python to predict inventory age to be able to manage a warehouse management problem, which is a big reason in the cost increases for a company that sells alcoholic beverages in a fcmg industry. After a deep research on warehouse and inventory management topics, it is estimetad that commonly used algorithms which are, Long-Short Term Memory (LSTM) and Recurrent Neural Networks (RNN) are implemented for a inventory age prediction data that is obtained. Echo State Network (ESN) algorithm is generally used for prediction and time series forecasting. Due to the ESN's complex structure it is foreseen that Echo State Network is a suitable algorithm for this problem in regard of commonly used methods poor outcomes.

Necesarry data selection and evaluation techniques like correlation matrix are used. To better understand which data can cause deviations and avoid overfitting or underfitting, data visualization is used for selected features by histogram, density graph's, statistical analysis matrix.

The machine learning algorithms are used for understanding complexity of the problem. Linear Regression, Polynomial Regression with Ridge Regression and Support Vector Machine models are created. Altough, it has seen that prediction score for linear regression and SVM models are very poor, Polynomial Regression with Ridge Regression prediction score is considerably sufficient with the score of 0.594 after inspecting the results.

Since machine learning model results did not live up to required levels, the deep learning models are created for Multilayer Perceptron (MLP), Recurrent Neural Networks (RNN), Long-Short Term Memory (LSTM) and Echo State Networks (ESN). Then, this models and results are compared by their R2 scores. Comparison of the results show that Echo State Network is the best model for this problem.

The contributions of this thesis can be specified as follows:

- Data collection is made with a great severity considering it is a criticially important parf of the problem. Due to the fact that data needs to be relevant,

it requires a balanced distribution, high quantity and should not include outliers.

- Feature selection and data evaluation is done by observing the correlation matrix.
- Statistical analysis is applied to acquire important values like mean, min, max and variance.
- Three Machine Learning (Linear, Polynomial with Ridge, SVM) models are constructed for understanding complexity of the problem.
- Four Deep Learning algorithms (MLP, RNN, LSTM, ESN) are modeled for better prediction results.
- The results are compared for deciding the best model for this problem and deficincies are observed.

This thesis can contribute to future research and thesises. Inventory Age has not been the subject of many studies. For future research, it is suggested that ESN is a effective way of predicting inventory age. Also, deep reinforcement learning algorithms are used for inventory control problems [41] [42] [43]. Our suggestion is that deep reinforcement learning algortihms can be used in further studies for inventory management prediction problems.

RESOURCES

[1] Short-term stock price prediction based on echo state networks Xiaowei Lin, Zehong Yang, Yixu Song, 2008.

[2] Machine learning models for biomass energy content prediction: A correlation-based optimal feature selection approach, Usman Alhaji Dodo, Evans Chinemezu Ashigwuike, Sani Isah Abba, 2022.

[3] Multi-label feature selection based on label correlations and feature Redundancy, Yuling Fan, Baihua Chen, Weiqin Huang, Jinghua Liu, Wei Weng, Weiyao Lan, 2022.

[4] New ridge regression, artificial neural networks and support vector machine for wind speed prediction, Yun Zheng, Yisu Ge, Sami Muhsen, Shifeng Wang, Dalia H. Elkamchouchi, Elimam Ali, H. Elhosiny Ali, 2023.

[5] Parallel SVM model for forest fire prediction, Kajol R Singh, K.P. Neethu, K Madhurekaa, A Harita, Pushpa Mohan, 2021.

[6] Prediction of new active cases of coronavirus disease (COVID-19) pandemic using multiple linear regression model Rath, Alakananda Tripathy, Alok Ranjan Tripathy A.C. P Fernandes, A. R Fonseca, F.A.L. Pacheco, L.F. Sanches Fernandes, 2020.

[7] A hybrid mental health prediction model using Support Vector Machine, Multilayer Perceptron, and Random Forest algorithms, E. Syed Mohamed, Tawseef Ahmad Naqishbandi, Syed Ahmad Chan Bukhari, Insha Rauf, Vilas Sawrikar, Arshad Hussain, 2023.

[8] An intelligent bankruptcy prediction model using a multilayer perceptron, Raffael Förch Brenes, Arne Johannssen, Nataliya Chukhrova, 2022.

[9] Applications of Artifcial Intelligence in Inventory Management: A Systematic Review of the Literature, Özge Albayrak Ünal, Burak Erkayman, Bilal Usanmaz, 2023.

[10] A Deep Learning-Based Inventory Management and Demand Prediction Optimization Method for Anomaly Detection, Chuning Deng and Yongji Liu, 2021.

[11] Stock Prıce Predıctıon Usıng LSTM, RNN And CNN-Sliding Wındow Model, Sreelekshmy Selvin, Vinayakumar R, Gopalakrishnan E.A, Vijay Krishna Menon, Soman K.P, 2017.

[12] Data-driven prediction model of indoor air quality in an underground space, Min Han Kim, Yong Su Kim, JungJin Lim, Jeong Tai Kim, Su Whan Sung & ChangKyoo Yoo, 2010.

[13] Application of ESN prediction model based on compressed sensing in stock market, Hao Zhang a, Mingwen Zheng a, Yanping Zhang a, Xiao Yu b, Wenchao Li a, Hui Gao, 2021.

[14] Research on predictive control of energy saving for central heating based on echo state network, Li Liu, Jun Liu, Zhifeng Chen, Zhongyang Jiang, Ming Pang, Yanshu Miao, 2023.

[15] Echo State Networks: Novel reservoir selection and hyperparameter optimization model for time series forecasting, Cesar H. Valencia, Marley M.B.R. Vellasco, Karla Figueiredo, 2023.

[16] A novel approach based on integration of convolutional neural networks and echo state network for daily electricity demand prediction, Sujan Ghimire, Thong Nguyen-Huy, Mohanad S. AL-Musaylh, Ravinesh C. Deo, David Casillas-Pérez, Sancho Salcedo-Sanz, 2023.

[17] Inventory write-down prediction for semiconductor manufacturing considering inventory age, accounting principle, and product structure with real settings, Jei-Zheng Wu, 2013.

[18] “A Study of Inventory Management Case Study”, Journal of Advanced Research in Dynamical & Control Systems, Nazar Sohail, Tariq Hussain Sheikh, Vol. 10, 10-Special Issue, 2018, pp. 1176-1190.

[19] "Pengantar Akutansi Konsep dan Tekhnik Penyusunan Laporan Keuangan," Rudianto, p. 222, 2012.

[20] Higher supply chain security with lower cost: lessons from total quality management, Lee, H., & Whang, S. Omega, 33(2), 143-144, 2005.

[21] An adaptive order quantity forecasting model for perishable products in online retailing. Mathematical Problems in Engineering, Zhang, X., Dong, X., & Wang, M., 2014.

[22] Machine learning for inventory management: a review. International Journal of Production Research, Srinivasan, R., & Mukherjee, A. 56(1-2), 233-265, 2018.

[23] A comprehensive review of estimation methods for demand management. Computers & Industrial Engineering, Fu, J., & Zhang, G. 86, 78-89, 2015.

[24] Linear Regression Using R- An Introductıon To Data Modeling Second Edition Davıd J. Lilja, Greta M. Linse, 2016.

[25] Introduction to Machine Learning Algorithms: Linear Regression, by Rohith Gandhi, 2018.

[26] Polynomial Regression Model of Making Cost Prediction In Mixed Cost Analysis, Isaac, O. Ajao, 2012.

[27] Complete Guide On Linear Regression Vs. Polynomial Regression With Implementation In Python, DataDrivenInvestor, 2023.

[28] Ridge ve Lasso Regresyonu: Temel matematiği ve Python uygulamasıyla, Medium, 2023.

[29] Understanding Machine Learning From Theory To Algorithms, Shai Shalev-Shwartz, Jerusalem Shai Ben-David, 2013.

[30] New Sequential and Parallel Support Vector Machine with Grey Wolf Optimizer for Breast Cancer Diagnosis Elsayed Badr, Sultan Almotairi, Mustafa Abdul Salam, Hagar Ahmed, 2021.

[31] "Neural Networks: A Comprehensive Foundation" by Simon Haykin, Chapter 1, Section 1.6.1, titled "The Multilayer Perceptron," page 16, 1994.

[32] "Neural Networks for Pattern Recognition" by Christopher M. Bishop, 1995.

[33] "Deep Learning" by Yoshua Bengio, Ian Goodfellow, and Aaron Courville, Chapter 10, 2015.

[34] “What are recurrent neural networks?”, IBM Watsons Studio.

[35] Long Short-term Memory, Hochreiter & Schmidhuber, 1997.

[36] Learning to Forget: Continual Prediction with LSTM. Neural Computation, Gers, F. A., Schmidhuber, J., & Cummins, F. 12(10), 2451–2471, 2000.

[37] Laplacian Echo State Network for Multivariate Time Series Prediction Min Han, Senior Member, IEEE, and Meiling Xu.

[38] “Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication,” H. Jaeger and H. Haas, Science, vol. 304, no. 5667, pp. 78–80, Apr. 2004.

[39] Chaotic Time Series Prediction Based on a Novel Robust Echo State Network, Decai Li, Min Han, Senior Member, IEEE, and Jun Wang, Fellow, IEEE.

[40] Short-term stock price prediction based on echo state networks, Xiaowei Lin, Zehong Yang, Yixu Song, 2009.

[41] Deep Reinforcement Learning for Inventory Control: a Roadmap, Robert Boute, Joren Gijsbrechts, Willem van Jaarsveld, Nathalie Vanvuchelen, 2022.

[42] Reward shaping to improve the performance of deep reinforcement learning in inventory management De Moor, B. J., Gijsbrechts, J., & Boute, R. N. 2021.

[43] Inventory management of new products in retailers using model-based deep reinforcement learning, Tsukasa Demizu, Yusuke Fukazawa, Hiroshi Morita, 2023.

RESUME

PERSONAL INFORMATION

Name Surname	: Melis KAMACIOĞLU
Date and Location of Birth	: 14.02.2000/TURKEY
Foreign Language	: English, German, Korean
E-mail	: meliskamacioglu@gmail.com

EDUCATION

Degree	**Field**	**School/University**	**Year of Graduation**
Bachelor	Industrial Engineering	Yildiz Technical University	2024
High School	-	Rize TOBB Science High School	2018

JOB EXPERIENCE

Year	**Company**	**Position**
2022-2023	BSH Home Appliances	IT Project Student
2021-2022	Smartin Information Technologies	HR Intern
2021	Bahadır Surgical Instruments	Production Planning Intern

PERSONAL INFORMATION

Name Surname : Özgür GÜMÜŞ

Date and Location of Birth : 18.10.1999/TURKEY

Foreign Language : English, German

E-mail : ozgurgumus1344@gmail.com

EDUCATION

Degree	Field	School/University	Year of Graduation
Bachelor	Industrial Engineering	Yildiz Technical University	2023
High School	-	Sakarya Anatolian High School	2017

JOB EXPERIENCE

Year	Company	Position
2023	Şişecam	Project Intern
2021	Yapı Kredi	Jr. Branch Intern
2021	Karmetal San. Tic. A.Ş	Production Planning Intern