SALES FORECASTING USING MACHINE LEARNING

A PROJECT REPORT

Submitted By

SHARAD GUPTA
(2100290140122)

YASH RAJ SINGH
(2100290140154)

Submitted in partial fulfillment of the Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Dr. Akash Rajak
Professor
KIET Group of Institutions

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Delhi-NCR,
GhaziabadUttar Pradesh- 201206

(FEBRUARY 2023)

CERTIFICATE

Certified that **Sharad Gupta (2100290140122), Yash Raj Singh (2100290140154)** have carried out the project work having "**Sales Forecasting Using Machine Learning**" for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Sharad Gupta(2100290140122)
Yash Raj Singh (2100290140154)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Dr. Akash Rajak

Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Signature of Internal Examiner **Signature of External Examiner**

Dr. Arun Tripathi
Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

Background: Sales forecasting is an important field in the food sector, and it has recently got immense popularity to boost market operations and productivity due to new technologies. The industry has traditionally focused on a conventional statistical model but in the recent years, Machine Learning techniques have received more attention.

Objectives: This thesis will help to identify the critical features that influence sales and also an experiment is performed to find the best suitable algorithm for sales forecasting.

Methods: Machine Learning Algorithms such as Simple Linear Regression, Gradient Boosting Regression, K-Nearest Neighbor, and Random Forest Regression were considered in thisthesis, which they expected to perform well on the issues. An experiment is carried out to determine the efficiency of the algorithms.

Results: Algorithms such as Simple Linear Regression, Gradient Boosting Regression, K-Nearest Neighbor, and Random Forest Regression are commonly known for performing better than others, this has been clearly shown that Random Forest Regression is the most appropriate algorithm compared to the others.

Conclusions: The Random Forest Regression algorithm performed well after doing all the study when compared with other algorithms. Hence the Random Forest Regression is considered as the best suitable algorithm for forecasting product sales.

ACKNOWLEDGEMENT

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, **Dr. Akash Rajak** for his guidance, help and encouragement throughout my researchwork. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Arun Kumar Tripathi, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion ofthis work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Sharad Gupta(2100290140122)

Yash Raj Singh(2100290140154)

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

Earlier companies used to produce goods without considering the number of sales and demand. For any manufacturer to determine whether to increase or decrease the production of several units, data regarding the demand for products on the market is required. Companies can face losses if they fail to consider these values while competing on the market. Different companies choose specific criteria to determine their demand and sales [1].

In today's highly competitive environment and ever-changing consumer landscape, accurate and timely forecasting of future revenue, also known as revenue forecasting, or sales forecasting, can offer valuable insight to companies engaged in the manufacture, distribution or retail of goods[2]. Short-term forecasts primarily help with production planning and stock management, while long-term forecasts candeal with business growth and decision-making[1].

Sales forecasting is particularly important in the industries because of the limited shelf-life of many of the goods, which leads to a loss of income in both shortage and surplus situations. Too many orders lead to a shortage of products and still too few orders lead to a lack of opportunity. Therefore, competition in the food market is continuously fluctuating due to factors such as pricing, advertisement, increasing demand from the customers[3].

Managers usually make sales predictions randomly. Professional managers, however, become hard to find and not always available (e.g., they can get sick or leave). Sales predictions can be assisted by computer systems that can play the qualified managers' role when they are not available or allow them to make the right decision by providing potential sales predictions. One way of implementing such a method is to try and model the professional managers' skills inside a computer program[4].

Alternatively, the abundance of sales data and related information can be used through Machine Learning techniques to automatically develop accurate sales predictive models. This approach is much simpler. It is not prejudiced by a single sales manager's particularities and is flexible, which means it can adapt to data changes. It has, however, the potential to overestimate the accuracy of the prediction of a human expert, which is normally incomplete. For example, once companies used to produce the products without taking into consideration the number of sales and demand as they faced several problems. Since they don't know how much to sell, forany manufacturer to decide whether to increase or decrease the number of units, dataregarding the consumer demand for products is essential. If companies do not con- sider these principles when competing in the market, they will face losses. Different companies choose different parameters to determine their market and sales.

There are several ways of forecasting sales in which companies have previously focused on various statistical models such as time series and linear regression, feature engineering and random forest models to obtain future sales and demand prediction. Time series contains data points that are stored over a fixed period and are used to forecast the future. Time series is a collection of data points which are collected in period at sequential, evenly spaced points. The most important components to analyze are patterns, seasonality, irregularity, cyclicity.

Linear regression is a mathematical tool used to forecast past values. It can help to determine the underlying trends and address cases involving overstated rates[5][6]. Feature engineering is the use of data on domain knowledge and the development of features to make predictive Machine Learning models more accurate. It makes for deeper data analysis and a more useful perspective[7]. A decision tree is a fundamental principle behind a model of random forests. The decision tree approach is a technique used in data mining to forecast and classify data. The decision tree approach does not provide any conceptual understanding of the issue itself. Random forest is the more sophisticated method that allows and merges many trees to make decisions. The random forest model results in more accurate forecasts by taking out an average of all individual tree decision predictions.

The entire data set is usually divided into two parts, namely the training data and the test data. Training data is a data that is used to train the model, and test data is the data used to evaluate the trained model. A classical approach is 80-20 split, stating that 80 percent of the data is used to train the model, and the remaining 20 percent of the data is used to test the model. But approaches like stratified K-fold cross-validation are known to provide good results. There were many cross-validation variants, such as simple k-folds, leave one out, stratified k-fold cross-validation, and so on[8][9].

1.0.1 Aims and Objectives

This thesis aims to develop a Machine Learning model that can predict the sales of products from different outlets. Several objectives were drawn to attain the goal:

Objectives:

- Converting data into an appropriate form using various preprocessing techniques for the implementation of Machine Learning algorithms.
- Finding critical features that will most influence sales of the product.
- To determine the appropriate Machine Learning algorithm for sales forecasting.
- Selecting various metrics to compare the performance of the applied Machine Learning algorithms.

1.0.2 Research Questions

Two research questions have been defined for this study to accomplish the aim. They are defined as follows:

RQ1:

What are the critical features that influence product sales?

Motivation:

The motivation of this research question is to find critical features in the data that can be useful while experimenting for RQ2 to build the Machine Learning model. This will help us reduce computational power and improves the quality of the results.

RQ2:

What is the best suitable algorithm for sales and demand prediction using Machine Learning techniques?

Motivation:

The critical features identified from RQ1 are used to develop the Machine Learning model using different algorithms. These models are compared by using various metrics such as accuracy score, mean absolute error, and max error to select the best fit model for the data.

1.1 Background

There are several methods for forecasting future demand for the goods and services a business provides. The forecasts are used for planning production and business activities, purchasing materials, inventory management, scheduling work hours, advertising, and often more across most industries. Traditional forecasting approaches were primarily focused on experienced employee opinions or statistical analysis of past data, but in recent years Machine Learning techniques have been implemented with great success in this field.

1.1.1 Data Mining

Data mining is described as a process for extracting usable data from a larger collection of raw data using statistical, artificial intelligence, Machine Learning and pattern recognition methods[10][11]. Data Mining is increasingly seen as a step ina systematic and iterative process of knowledge discovery, in which automated pat- tern recognition methods are combined with expert knowledge of the analyst. This process is called the Knowledge Discovery in Databases (KDD) process[12].

1.1.2 Machine Learning

Machine Learning is the area of study which enables machines to learn without being explicitly programmed[13]. Machine Learning is defined as the computer program learns from experience E with respect to some class of tasks T and performance measure P when its performance at tasks in T, as measured by P, strengthens with

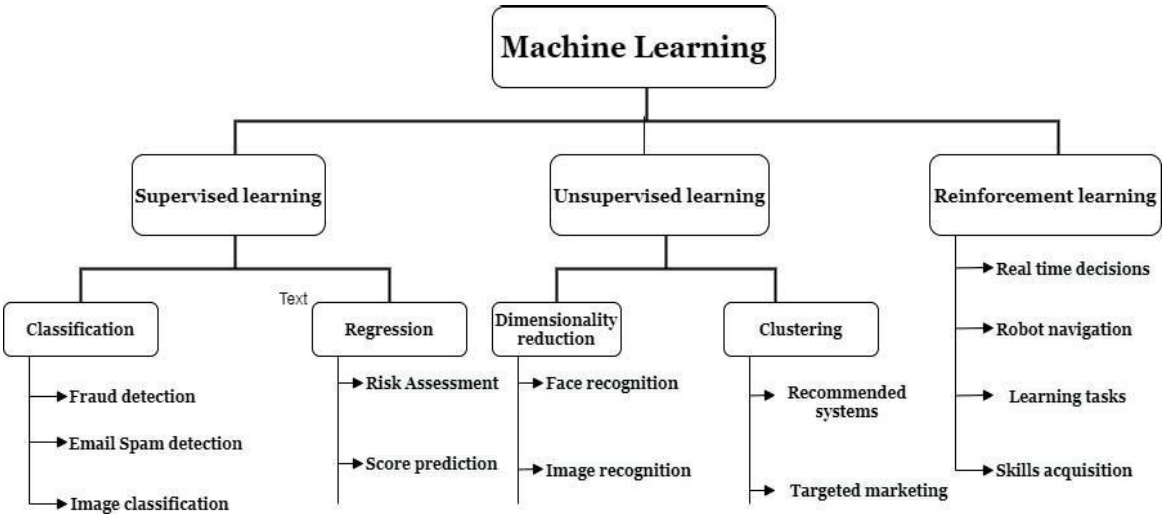

Figure 1.1: Types of Machine Learning

experience E[14]. In general, Machine Learning is a program that can manage various tasks by analyzing and exploring data[15].

Common Machine Learning applications such as email spam detection, credit card fraud, stock predictions, smart assistants, product recommendations, self-driving cars, sentiment analysis, etc.

Supervised Learning:

The most popular model for performing Machine Learning processes is supervised learning. It is commonly used for data where the mapping between input-output data is accurate. Supervised learning is the subset of Machine Learning which concentrates on learning a model of classification or regression, that is, learning from labeled test data[15].

Unsupervised Learning:

The data is not explicitly labeled into different classes in the case of unsupervised learning that is there is only unlabeled data. By identifying implicit patterns the model can learn from the data. Unsupervised Learning categorizes the densities, structures, related segments, and other similar properties based on the data[16].

Reinforcement Learning:

Reinforcement Learning is a sub-field of Machine Learning. In a given scenario, it is about taking appropriate action to optimize reward. Various algorithms and computers are employed to determine the best possible action or path it will follow in a specific scenario. Reinforcement learning varies with supervised learning in such a way that the training data has the answer key with it in supervised learning such that the model is trained with the correct response itself while in reinforcement learning there will be no response but the reinforcement agent determines how to

execute the task. It is required to learn from its experience, in the absence of training data[15].

1.1.3 Machine Learning Algorithms:

Forecasting means predicting events of the future, typically based on previous records. For a long time, statistical models were commonly used for the conducting of predictions. The role of generalization in Machine Learning has been considered. In the case when a new product or store is introduced, this effect could be used to make sales predictions because there is a limited amount of historical data for a particular time series[17]. In this thesis we have used supervised learning algorithms such as Support Vector Machines, Random Forest Regression, Gradient Boosting, and Sim- ple Linear Regression. These can make it easier to find better outcomes comparedto traditional analytical techniques of time series[18].

Simple Linear Regression

Simple linear regression is useful for defining a relationship between two continuous variables. One is an indicator or independent variable and another is an answer or dependent variable. It looks for a statistical relationship, but not a deterministic one. The relationship between the two variables is said to be deterministic if one variable can be precisely represented by the other[19]. For example, it is possible to correctly forecast Fahrenheit by using temperature in degree Celsius. The mathemat- ical equation is not sufficient to assess the association between the two variables. For example, the relationship between weight and height. The Equation for the Simple linear regression is:

Extreme Gradient Boosting Regression

Extreme Gradient boosting is some kind of enhancement in Machine Learning. It is basedon the premise that, when combined with previous ones, the best possible current iteration will minimize the maximum prediction error. The key idea for this next iteration is to set the target outcomes to minimize the error.

One of the most successful Machine Learning models for predictive analytics is the Gradient Boosted Regression Trees (GBRT) model (also called Gradient Boosted Machine or GBM), which makes it an industrial workhorse for Machine Learning.The Boosted Trees Model is a type of additive model that combines decisions from a sequence of base models to make predictions[20]. One can write this class of models more formally, as:

K-Nearest Neighbor

K-Nearest Neighbor is one of the simplest machine learning algorithms based on Supervised Learning technique . KNN algorithm stores all the available data point based on the similarity . This means when new data appears then it can be easily classified into a well suit category by using K-NN algorithm.

Random Forest Regression

Random Forest is one of the most powerful Machine Learning frameworks for predictive analytics. A random forest method is a type of discrete structure that allows predictions by integrating decisions from a series of simple models[22]. More formally, this subset of models can be written as:

1.1.4 Selection of Machine Learning Algorithms

For every problem, choosing an algorithm is not a trivial decision. There is no proper algorithm that works for any problem, but few algorithms are widely recognized for performing the algorithms better than others in some cases. One can not assume the more accuracy from the algorithms for all types of data, accuracy will differ from data to data. In this thesis Machine Learning Algorithms such as Simple Linear Regression, Gradient Boosting Regression, Support Vector Regression, and Random Forest Regression were considered in which they expected to perform well on the issues.

1.1.5 Selection of Performance Metrics

To identify the appropriate algorithm one needs to evaluate the results and then we can predict it. In this case, the accuracy score would play a crucial role while measuring the performance of the algorithm. For calculating the average magnitude of errors mean absolute error metric will be used in this study. In real-time data there might be a chance of worst-case error between the actual value and predicted value in this particular scenario max error is used.

Chapter 2

Related Work

Previously a lot of sales and demand forecasting work was performed using Machine Learning. Most of the work in this research will concentrate on the sales of food items. Due to the importance of forecasting in various fields, there are so many different types of approaches taken previously, some of the methods such as Machine Learning models, hybrid models, and statistical models. To handle this work, some of the statistical methods such as auto regressive moving average (ARMA) and auto regressive integrated moving average (ARIMA) will be helpful[4].

İrem İşlek and Şule Gündüz Öğüdücü experimented with the use of bipartisan graphic clusters that clustered different warehouses according to the sales behavior. They addressed the application by applying the Bayesian network algorithm in whichthey managed to produce the enhanced forecasting experience[23].

Grigorios Tsoumakas had used Machine Learning techniques to perform a surveyon the forecasting of food sales. They had addressed data analyst design decisions such as temporal granularity, output variable, and input variables in this survey[4].In this paper the authors experimented by taking the point of sale (POS) as internal data and even external data by considering different environments to enhance the efficiency of demand forecasting. They considered different Machine Learning algo-rithms such as Boosted Decision Tree Regression, Bayesian Linear Regression, and Decision Forest Regression for evaluation[24].

The paper's authors had researched interestingly about customers coming to the restaurants using Random Forests, k-nearest neighbor, and XGBoost. They chose two real-world data sets from different booking sites and also made different input variables from restaurant features. The results have shown that XGBoost is the most appropriate model for the dataset[25].

Holmberg and Halldén had observed that regular restaurant sales to be influencedby the weather. They considered two Machine Learning algorithms as XGBoost and neural network, and the results showed that the XGBoost algorithm is more accurate than the other algorithm, and they also found that they had improvedtheir model performance by 2-4 percentage points by taking weather factors into consideration. To improve accuracy, they had considered numerous variables such as date characteristics, sales history, and weather factors[26].

Most of the recent studies focused on sales modeling without considering the relationship between the training and testing data, they used training data directly. This causes many errors which lead to a reduction in accuracy. Recent studies have suggested clustering techniques to separate the entire forecasting data into

several clusters of predictable data before designing predictable models to minimize computational time and achieve effective evaluating performance[27].

In particular, Support Vector Machine(SVM) had been applied to demand forecasting. Garcia et al. (2012), in their study, proposed an intelligent model that relies on supporting vector machines to deal with issues relating to the allocation and revelation of new models. Kandananond (2012) showed that SVM surpassed Artificial Neural Networks in estimating demand for consumer goods[28].

Previously, most of the studies focused on considering the metrics as mean absolute error, mean squared error, median absolute error, and k-fold cross validation is used for training and testing data. Metrics like max error, accuracy, and mean absolute error are considered in this research. In this study stratified K-fold cross- validation technique is used for training and testing to increase the efficiency of the results. In this study a suitable algorithm is chosen for sales forecasting.

Chapter 3

Method

In this thesis research questions are answered by using research methods. Research aspects for this work are examined by the execution of the experiments.

3.1 Experiment

An experiment is chosen for the first research question i.e. correlation. Each data attribute can be selected by applying feature selection methods like data correlation and which will make the predictable attributes more accurate. This will reduce a lot of strain on the Machine Learning model during pre-processing and cleansing the data. For the second research question an experiment is chosen because the experiments provide control over factors and a deeper understanding of many common research techniques such as a case study or survey[29]. One can describe the procedure followed in this experiment as follows:

- Extracting the data required for the sales.
- Applying specified Machine Learning (supervised) algorithms.
- The performance of the output can be enhanced by comparing metrics such as accuracy score, mean absolute error and max error.
- Based on assessment tests, the best suitable algorithm can be selected.

3.2 Experimentation Environment

Python

Python is a commonly used high-level programming language, it was designed by Guido van Rossum which can be easy to interpret and read[30]. Python has specific functionality and is convenient to be used for both quantitative and analytical computational purposes. Data Science Python is popularly used and, as well as being a dynamic and open source language, is a top choice. Its massive libraries are also used to manipulate the data however for a beginner data analyst they are really simple to learn[31]. The python libraries used in this thesis are briefly described as follows:

NumPy

NumPy is a library that consists of multidimensional array objects and a set of array processing routines. NumPy is used along with SciPy and Matplotlib packages. This combination is used for technical computing. Mathematical and logical operations are performed with the help of NumPy[32].

Pandas

Pandas is a software library that is designed for manipulating the data and analysisin a python programming language. It is open-source which is released under the BSD license of three clauses. It is based on the Numpy package, and the DataFrame is its main data structure[33].

Matplotlib

Matplotlib is a module of Python used to plot the attractive Graphs. Visual representation in data science is a significant step. One can quickly understand how data is split by using visual representation. There are many libraries to representthe data, but the matplotlib is very widely known and easier to visualize[34].

SKlearn

Scikit-learn is a free python library. It features multiple clustering classification and regression algorithms including random forests, DBSCAN, k-means, gradient boosting, support vector machines, and gradient boosting which is programmed to interface with the NumPy and SciPy libraries[35].

Seaborn

Seaborn is a open-source python library that is used for statistical graphics. It offers a data set-oriented API to analyze relationships among different variables, as well as resources to select color palettes that truly in the data[36].

3.3 Data overview

In this thesis, there is labeled sales data from different items from different outlets that provide information such as item type, item price, outlet type, etc. These data were extracted from various sources and will be used to train and improve the modelfor Machine Learning. In the dataset being analyzed there are 8523 instances and12 attributes. The dataset has been properly divided into training and testing data that can be described in the sections below.

3.4 Feature Selection

There are various types of factors that can make the model of Machine Learning more effective on any given task. One of the methods of feature selection is data correlation

	Store	Date	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI	Unemployment	IsHoliday
8185	45	2013-06-28	76.05	3.639	4842.29	975.03	3.00	2449.97	3169.69	NaN	NaN	False
8186	45	2013-07-05	77.50	3.614	9090.48	2268.58	582.74	5797.47	1514.93	NaN	NaN	False
8187	45	2013-07-12	79.37	3.614	3789.94	1827.31	85.72	744.84	2150.36	NaN	NaN	False
8188	45	2013-07-19	82.84	3.737	2961.49	1047.07	204.19	363.00	1059.46	NaN	NaN	False
8189	45	2013-07-26	76.06	3.804	212.02	851.73	2.06	10.88	1864.57	NaN	NaN	False

Figure 3.1: Dataset description

```
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Store         8190 non-null   int64
 1   Date          8190 non-null   object
 2   Temperature   8190 non-null   float64
 3   Fuel_Price    8190 non-null   float64
 4   MarkDown1     8190 non-null   float64
 5   MarkDown2     8190 non-null   float64
 6   MarkDown3     8190 non-null   float64
 7   MarkDown4     8190 non-null   float64
 8   MarkDown5     8190 non-null   float64
 9   CPI           8190 non-null   float64
 10  Unemployment  8190 non-null   float64
 11  IsHoliday     8190 non-null   bool
dtypes: bool(1), float64(9), int64(1), object(1)
memory usage: 712.0+ KB
```

Figure 3.2: Dataset summary

Date	Weekly_Sales	Size	Temperature	Fuel_Price	CPI	Unemployment	IsHoliday	Year	Month	Week	...	Dept_93	Dept_94	Dept_95	Dept_96	Dep
2010-02-05	0.342576	0.630267	0.328495	0.050100	0.840500	0.508787	0	2010	2	5	...	0	0	0	0	
2010-02-05	0.257324	0.925525	0.345332	0.063126	0.003737	0.571016	0	2010	2	5	...	0	0	0	0	
2010-02-05	0.213756	0.318073	0.120065	0.158317	0.054008	0.744463	0	2010	2	5	...	0	0	0	0	
2010-02-05	0.043986	0.318073	0.120065	0.158317	0.054008	0.744463	0	2010	2	5	...	0	0	0	0	
2010-02-05	0.148717	0.318073	0.120065	0.158317	0.054008	0.744463	0	2010	2	5	...	0	0	0	0	

5 rows × 145 columns

Figure 3.3: Correlation values

which will have a major impact on the model's performance. This will reduce a lot of strain on the Machine Learning model during preprocessing and cleansing the data. The data attributes chosen for training the Machine Learning model would have a major impact on the efficiency of the model. Because of the irrelevant features that are presented, the model output will be reduced. The feature selection methodprovides an efficient way to remove data redundancy and irrelevant data that helpsto reduce computation time, improve accuracy, and also enhance understanding of themodel[37].

The selection of features plays a crucial role in classification and involves selecting a subset of features that reflect the complete attributes that currently exist. Feature selection techniques are intended to improve classification efficiency by selecting the essential features from the data sets according to particular algorithms.

3.4.1 Data Correlation Method

Data correlation is a method that helps to predict one attribute from another at- tribute and is used as a basic quantity in many modeling techniques. If one feature increases, the correlation will be positive, so the other feature increases as well and negative if one feature increases there will be a reduction in another. If there is no relation between any two attributes then it is said to be no correlation[38]. If thereis a linear relationship between the constant variables then the Pearson correlation coefficient is used. If there is a non-linear relation between the constant variables then the Spearman correlation coefficient is used.

Since the considered data set is linear so the Pearson correlation coefficient is used for the selection of features in this study. This correlation for all the attributesis shown in figure 3.4. To improve the efficiency of the Machine Learning model, the attributes that have negative correlations were removed. It is a statistic measuringthe linear correlation of two variables X and Y. It has a value between +1 and 1,where 1 is a linear positive correlation, 0 is not a linear correlation and 1 is a linear negative correlation[39].

The motivation for considering the correlation is when people know a score on one measure, they can make a prediction of another measure that is highly relatedto it more accurate. The more accurate the prediction, the stronger the relationship between the variables.

The heat map for correlation between non-numerical attributes is plotted as follows:

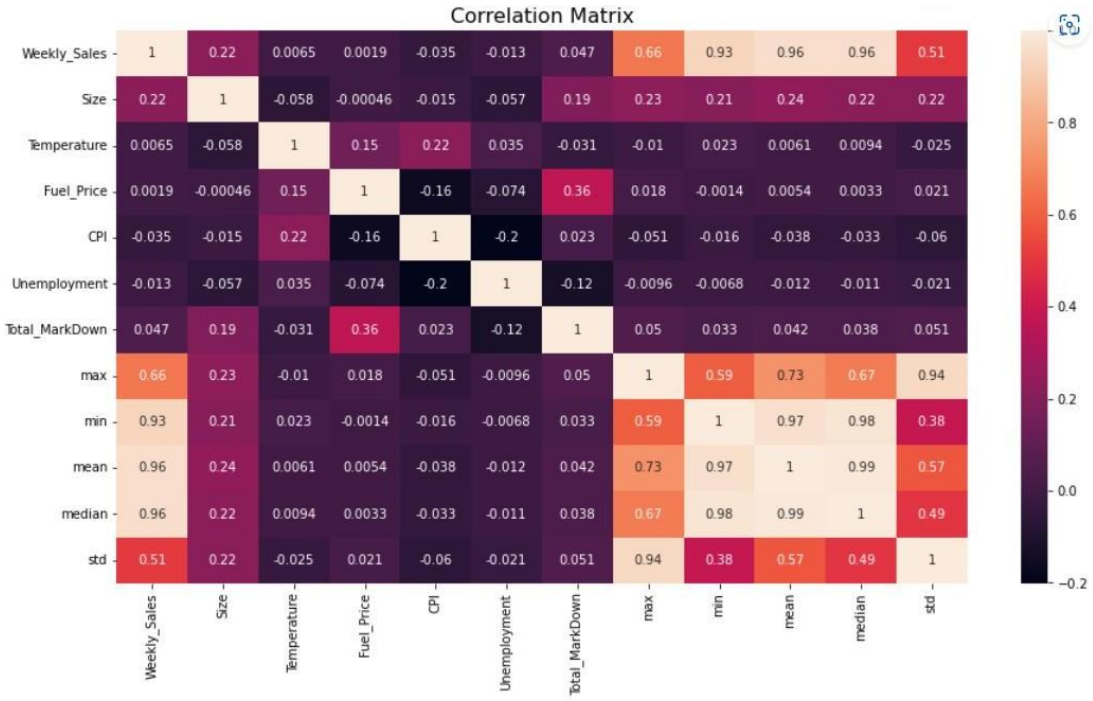

Figure 3.4: Heat map

3.5 Feature Importance

Feature Importance refers to a class of approaches for assigning values to input features to a predictive model which determines the relative significance of each factor while forecasting[40].

Feature importance scores provide overview into the model. Most significant scores are determined using a prediction approach that was fitted to the dataset. Inspecting the score of importance gives insight into that particular model and what features are the most essential and least important to the model while making a prediction. This is a type of interpretation of the model that can be carried out for those models that encourage it.

Feature Importance can be used to enhance a predictive model. This can be accomplished by selecting those features to remove (lowest scores) or those features to retain, using the importance scores. This is a type of selection of features, and can simplify the modeling problem, accelerate the modeling process, and in certain cases improve model performance[40].

3.6 Data preprocessing

Before applying Machine Learning algorithms some of the missing values have been found which can impact the model's output so this should be handled. The 'item weight' and 'outlet size' attributes have 17 percent, and there is 28 percent of missing values. To make the dataset more efficient, these missing values will be replaced by the most promising values. There's more correlation between two of the different attributes with similar work. Removing one of the attributes will make the work better. The redundant values such as LF and reg provided in the attribute of itemfat content will be treated and these redundant values will be replaced accordingly. The least value for an 'item visibility' attribute is zero which makes no sense for the dataset.

3.6.1 Encoding Categorical Values

Categorical data contains label values that are considered nominal values. Each value has categories of different types. Besides, a few of the groups have a normal relationship with each other is known as natural ordering. The categorical data can be converted into numerical data to improve the efficiency of the Machine Learning model[41].

One Hot Encoding

One hot encoding is the method where the data is represented in binary format and included as a feature. It is one of the most common methods, comparing each level of the numerical variable with a fixed starting point. In this thesis for the data set that had taken, one hot encoding is used to represent categorical variables as binary vectors[41].

This approach results in a dummy variables trap because it is easy to predict the outcome of one variable with the support of the existing variables. This trap leads

Date	Store	Dept	Type
2012-10-26	19	31	A
2012-10-26	3	79	B
2012-10-26	19	46	A
2012-10-26	19	27	A
2012-10-26	45	98	B

Figure 3.5: Before One Hot Encoding

Date	Store_1	Store_2	Store_3	Store_4	Store_5	Store_6	Store_7	Store_8	Store_9	Store_10	...	Dept_93	Dept_94	Dept_95	Dept_96	Dept_97	Dept_98
2010-02-05	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
2010-02-05	0	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0
2010-02-05	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
2010-02-05	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
2010-02-05	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0

Figure 3.6: After One Hot Encoding

to a multicollinearity problem. It occurs when the independent features become dependent upon each other. To overcome the multicollinearity problem one of the dummy variables should be dropped. The following figure represents the before and after one hot coding.

3.6.2 Stratified K-fold Cross-Validation

Cross-validation (CV) is a procedure of statistical analysis used to assess the effectiveness of a Machine Learning technique, as well as a re-sampling method used to validate an algorithm if there is insufficient data[42].

Stratification is the process of rearranging the data to ensure each fold is a good representative of the whole. Data splitting into folds may be controlled by criteria such as ensuring where each fold has the same ratio of outcomes with a given cate- goricalvalue, such as the class outcome value. This process is called stratified k-fold cross-validation[43]. Common techniques of cross-validation include K-fold cross- validation, Stratified K-fold cross-validation, and cross-validation leave-one-out.

The motivation behind the 10-fold stratified cross-validation is that the estimator has a lower variance than a single hold-out set estimation method which could be very essential if there is a limited amount of data. There will be plenty of variance inthe results estimate for various data samples, or for specific data partitions to create training and test sets. The 10-fold stratified cross-validation removes this variance by comparing more than 10 separate partitions, thereby making the performance estimate less sensitive to data partitioning.

3.7 Performance Metrics

Several metrics can be used while evaluating how well a model is performing. It is necessary to understand how each metric measures to select the evaluation metric to better assess the model. This thesis main objective was to compare the performance of Machine Learning techniques by evaluating all of these performance metrics such as Accuracy score, Mean Absolute Error, and Max error.

3.7.1 Accuracy score

Accuracy is known as the ratio of a several correct predictions(both true positives and true negatives) to the total number of data points[44].

$$AccuracyScore = \frac{FN + FP}{N}$$

Where FN is false negative, FP is false positive and N is total number of predictions.

3.7.2 Max Error

The function max error measures the maximum standard errors, a metric representing a worse-case error between the expected value and the actual value. Max error

would be 0 on the test set in a properly fitted single-output regression analysis, and while this would be extremely impossible in the modern world, this measurement indicates the amount of error the model has when it was placed in[45].

$$MaxError(y, x) = max(|y_i - x_i|)$$

Where Yi describes the actual values, Xi describes the expected values.

3.7.3 Mean Absolute Error

Mean Absolute Error is a process performance measure that is used for regression models. A model's mean absolute error concerning a test data set is the average ofthe actual values on all instances in the test set of the specific prediction errors. For instance, every predictive error is the difference between the predicted value and the actualvalue. Mean Absolute Error is one of several metrics for summing up andmeasuring the Machine Learning model's performance[46].

$$MAE(y, x) = \left(\frac{1}{n_{samples}}\right) \sum_{i=0}^{n_{samples}-1} |y_i - x_i|$$

Where Yi describes the actual values, Xi describes the expected values.

Chapter 4

Results

Simple Linear Regressor, Xtreme Gradient Boosting Regressor, Random Forest Regressor and K-Nearest Neighbor are trained with the set of data using a 10-fold stratified cross-validation approach that dynamically selected the training and testing with fixedproportion each time and the efficiency was calculated using max error, mean absoluteerror and accuracy metrics.

4.1 Simple Linear Regressor

The Values present in figure 4.1 are generated by Simple Linear Regressor In figure 4.2 , Real Values and Predicted Values are generated by Simple Linear Regressor , figure 4.3 , represents the Graph on the basis of Real values and Predicted values , figure 4.4 shows the accuracy of Simple Linear Regressor.

The outcomes that Simple Linear Regressor obtains are as follows:

```
MAE 0.030244102384805
MSE 0.0035838384133818763
RMSE 0.059865168615663954
R2 0.919972025288203
```

Figure 4.1: Values generated by Simple Linear Regression

Date	Actual	Predicted
2011-08-05	0.335204	0.461618
2010-07-09	0.097150	0.131949
2011-07-01	0.003508	0.011631
2012-01-06	0.081150	0.072653
2011-08-26	0.006916	0.030492
...	...	...
2011-01-28	0.000034	-0.010030
2010-08-20	0.008975	0.013981
2010-11-26	0.026455	0.028503
2010-03-12	0.002350	0.008520
2010-02-12	0.256692	0.269139

74850 rows × 2 columns

Figure 4.2: Actual and Predicted Values generated by Simple Linear Regression

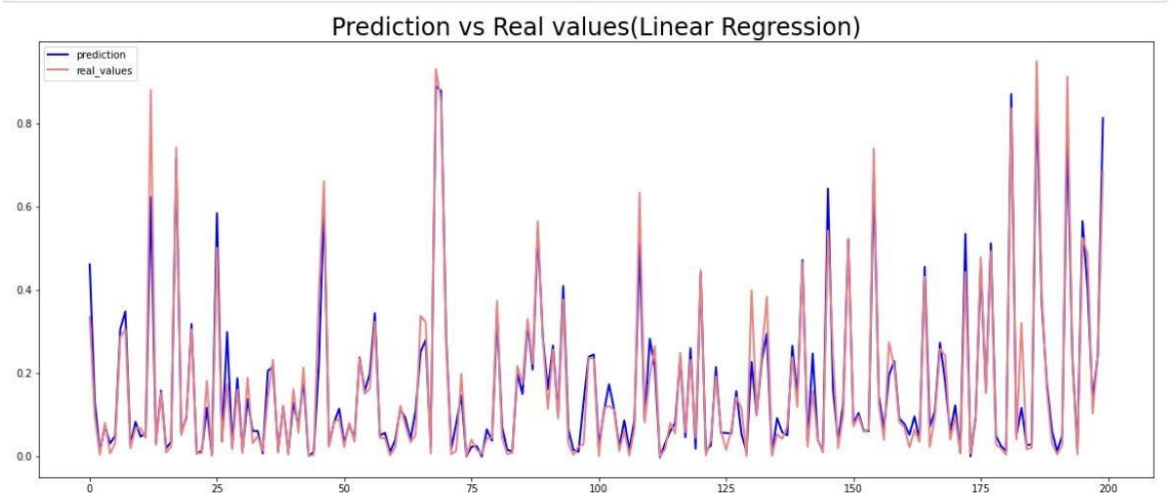

Figure 4.3: Graph for Real and Predicted Values generated Simple Linear Regressor

```
Linear Regressor Accuracy -  91.99697914094334
```

Figure 4.4: Accuracy generated by Simple Linear Regressor

4.2 Extreme Gradient Boosting Regressor

The Values present in figure 4.5 are generated by Extreme Gradient Boosting Regressor, In figure 4.6 , Real Values and Predicted Values are generated by Extreme Gradient Boosting Regressor , figure 4.7 , represents the Graph on the basis of Real values and Predicted values , figure 4.8 shows the accuracy of Extreme Gradient Boosting Regressor.

The outcomes that Gradient Boosting Regressor obtains are as follows:

Figure 4.5: Values generated by Extreme Gradient Boost Regressor

```
MAE 0.020195384508839235
MSE 0.0012656262397593718
RMSE 0.035575641101171625
R2 0.9717375296815178
```

Figure 4.6: Real and Predicted values by Gradient Boosting Regressor

Date	Actual	Predicted
2011-08-05	0.335204	0.404160
2010-07-09	0.097150	0.128982
2011-07-01	0.003508	0.004509
2012-01-06	0.081150	0.071191
2011-08-26	0.006916	0.008820
...	...	...
2011-01-28	0.000034	0.001186
2010-08-20	0.008975	0.011092
2010-11-26	0.026455	0.023844
2010-03-12	0.002350	-0.003059
2010-02-12	0.256692	0.277494

74850 rows × 2 columns

Figure 4.7: Accuracy Graph plot Extreme Gradient Boosting Regressor

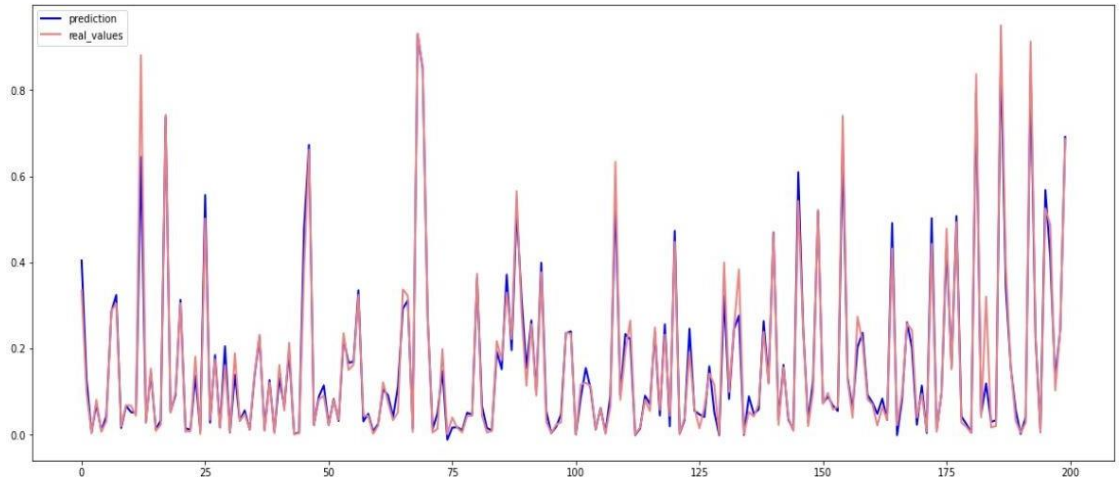

Figure 4.8: Accuracy Graph plot Extreme Gradient Boosting Regressor

```
XGBoost Regressor Accuracy -  97.17374724297191
```

4.3 K -Nearest Neighbor

The Values present in figure 4.9 are generated by K-Nearest Neighbor, In figure 4.10 , Real Values and Predicted Values are generated by K-Nearest Neighbor , figure 4.11 , represents the Graph on the basis of Real values and Predicted values , figure 4.12 shows the accuracy of K-Nearest Neighbor.

The outcomes that K-Nearest Neighbor obtains are as follows:

```
MAE 0.03346343602908097
MSE 0.0037285456933715765
RMSE 0.061061818621554145
R2 0.9169838361100898
```

Figure 4.9: Values Generated by K-Nearest Neighbor

Date	Actual	Predicted
2011-08-05	0.335204	0.491052
2010-07-09	0.097150	0.110391
2011-07-01	0.003508	0.005878
2012-01-06	0.081150	0.069880
2011-08-26	0.006916	0.012011
...	...	...
2011-01-28	0.000034	0.000137
2010-08-20	0.008975	0.004970
2010-11-26	0.026455	0.033908
2010-03-12	0.002350	0.000233
2010-02-12	0.256692	0.212626

74850 rows × 2 columns

Figure 4.10: Real and Predicted for K-Nearest Neighbor

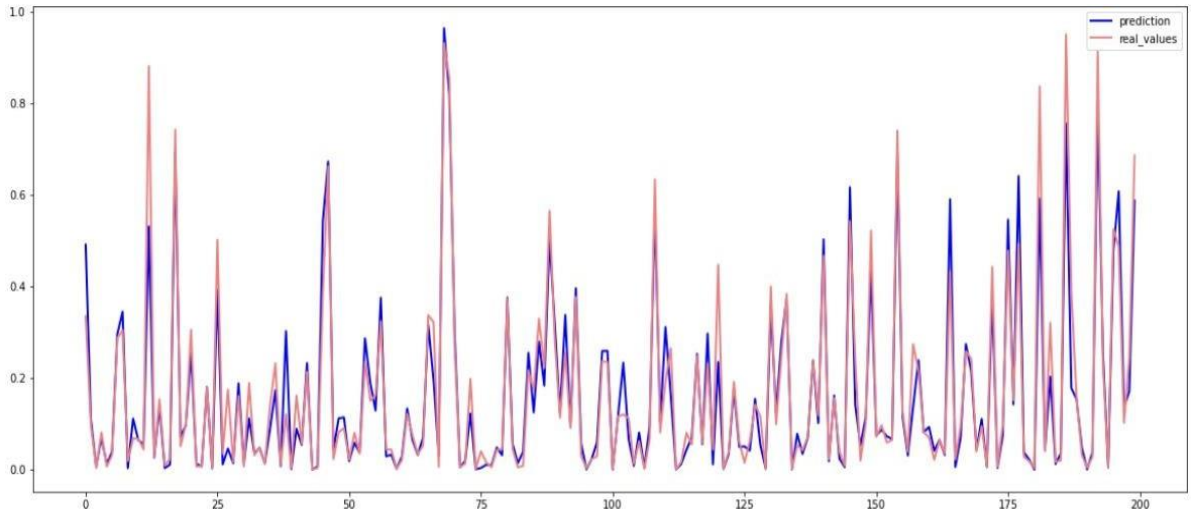

Figure 4.11: Real vs Predicted Graph for K-Nearest Neighbor

```
KNeigbhbors Regressor Accuracy -  91.67383528046942
```

Figure 4.12: Accuracy Generated by K-Nearset Neighbor

4.4 Random Forest Regressor

The Values present in figure 4.13 are generated by Random Forest Regressor In figure 4.14 , Real Values and Predicted Values are generated by Random Forest Regressor , figure 4.15 , represents the Graph on the basis of Real values and Predicted values , figure 4.16 shows the accuracy of Random Forest Regressor

The outcomes that Random Forest Regressor obtains are as follows:

```
MAE 0.015652788820879657
MSE 0.000989946656020751
RMSE 0.03146341774220898
R2 0.9778936449672184
```

Figure 4.13: Values generated by Random Forest Regressor

Date	Actual	Predicted
2011-08-05	0.335204	0.362592
2010-07-09	0.097150	0.109223
2011-07-01	0.003508	0.004925
2012-01-06	0.081150	0.062868
2011-08-26	0.006916	0.008590
...	...	...
2011-01-28	0.000034	0.000083
2010-08-20	0.008975	0.009193
2010-11-26	0.026455	0.024328
2010-03-12	0.002350	0.001866
2010-02-12	0.256692	0.254595

74850 rows × 2 columns

Figure 4.14: Real and Predicted values by Random Forest Regressor

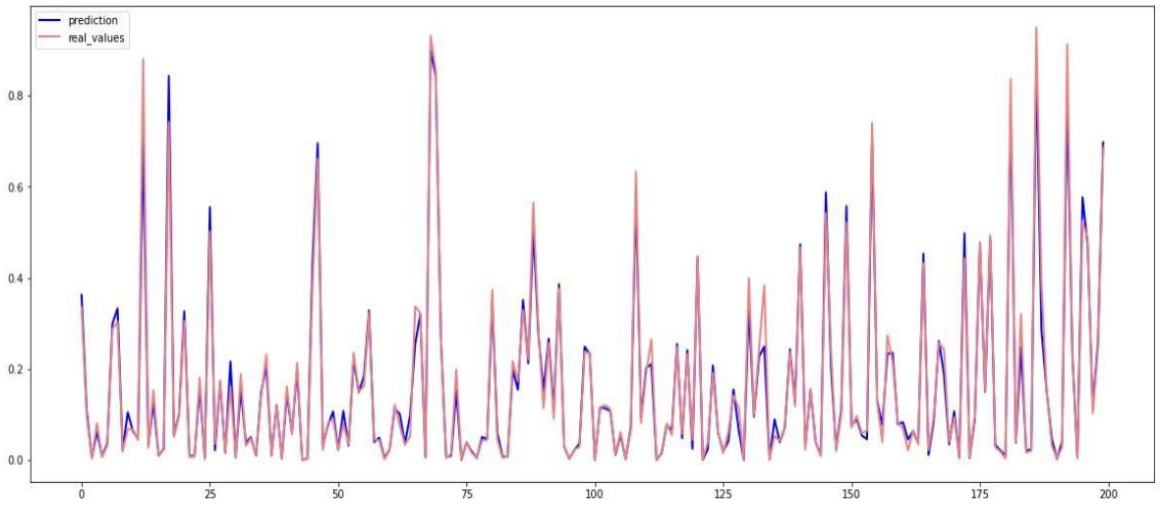

Figure 4.15: Real and Predicted values by Random Forest Regressor

```
Random Forest Regressor Accuracy -  97.78936357512521
```

Figure 4.15: Accuracy Generated by Random Forest Regressor

4.5 Feature Importance

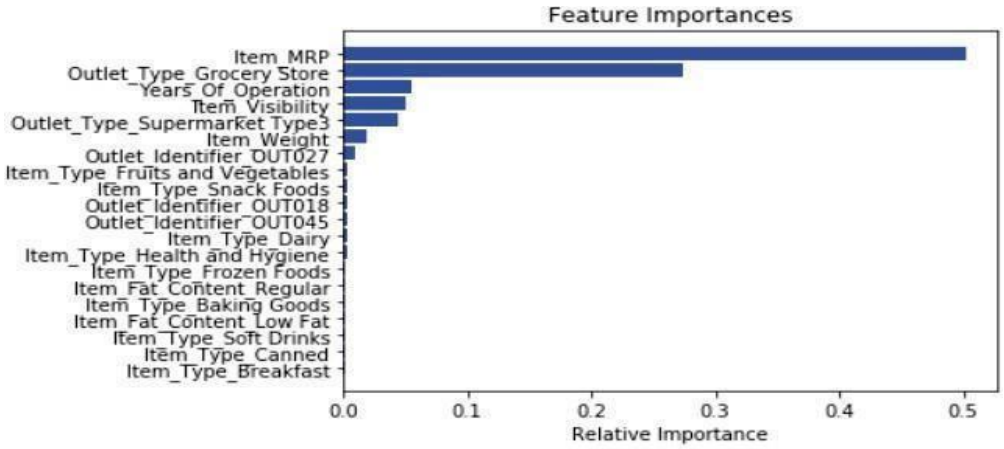

Figure 4.17: Feature Importance

Figure 4.17 shows that the feature importance of price of the products would depend primarily on the sales followed by the type of outlet and grocery store andthe rest of the features would not even close to these features. There will surely bea huge impact on sales forecasting with these features.

4.6 Evaluation Results

	model	accuracy
0	lr_acc	91.996979
1	rf_acc	97.789364
2	knn_acc	91.673835
3	xgb_acc	97.173747

Figure 4.18: Comparison of Evaluation Results

Table 4.1 shows the comparison of evaluation results where Random Forest Regression performed well with all the metrics accuracy score, Mean Absolute Error and Max Error. Random Forest Regression had the minimum error in predicting the sales when compared to the Simple Linear Regression, Gradient Boosting regression and K-Nearest Neighbor. Simple Linear Regression demonstrated the worst performance with the highest error in all the metrics. A simplified tabular form based on the results is created above.

5.1 Comparative analysis of Performance Metrics

In this section the average determined on the basis of ten iterations that were considered for all the metrics.

5.1.1 Average Accuracy Score

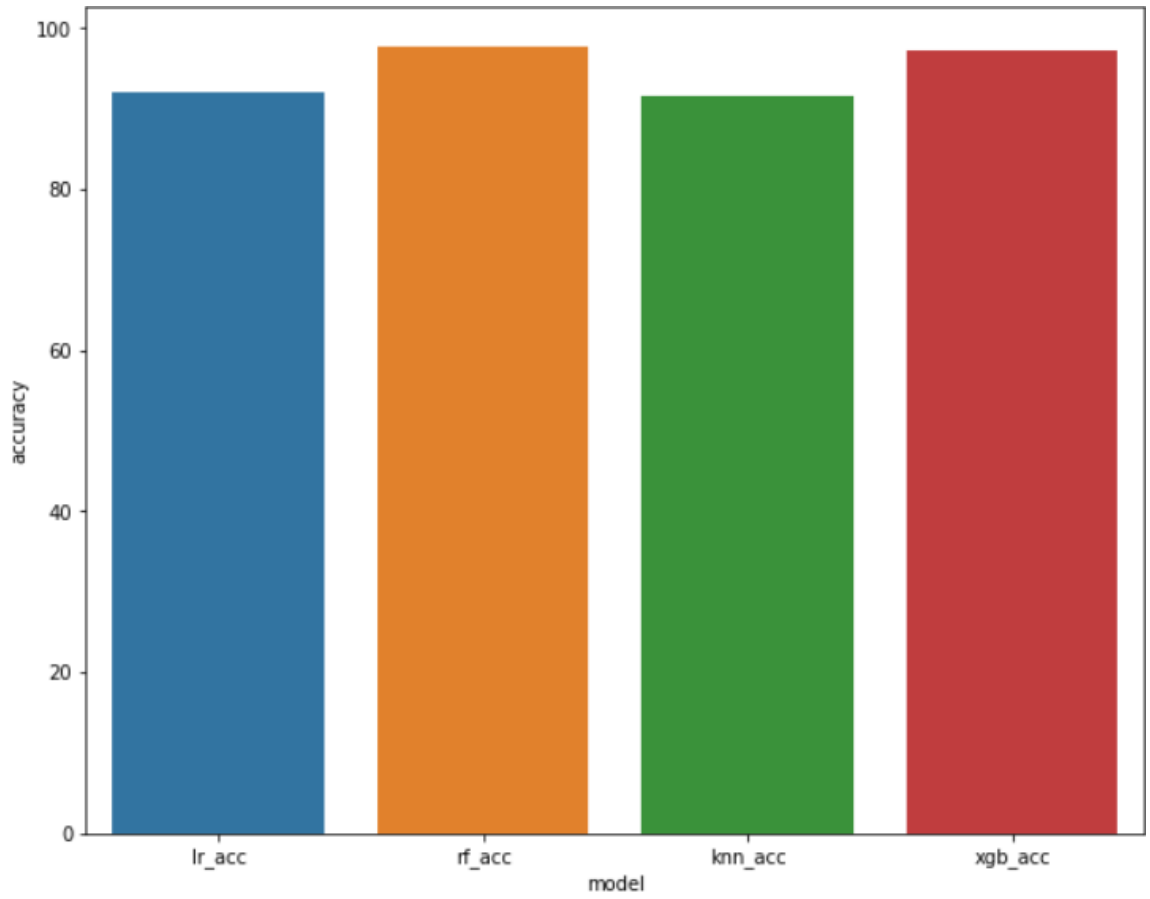

Figure 5.1 Accuracy score plot

Figure 5.1 shows the average accuracy score of the 10-fold stratified cross-validation obtained by the Simple Linear Regressor is 81.2 percent, followed by the Gradient Boosting Regressor with 86.27 percent accuracy score, then KNN with the 84.82 per-cent accuracy score and finally Random Forest Regressor with 87.72 percent accuracy

score. From figure 5.1, it can be seen that Random Forest Regressor is the best per- former with approximately 88 percent accuracy score compared to other methods, andSimple Linear Regressor is the poor performer with an accuracy score of 81.2 percent.

5.2 Discussion

RQ1: What are the critical features that influence product sales?

Answer:

It was clearly observed in figure 4.13 that the price of the products and followed by the type of outlet and grocery store will heavily influence the product sales.

RQ2:

What is the best suitable algorithm for sales and demand prediction using Machine Learning techniques?

Answer:

Random Forest Regression is the most appropriate algorithm for forecasting the product sales. When compared to the Simple Linear Regression, Gradient Boosting Regression and K-Nearest Neighbor, Random Forest Regression technique will produce the least error while predicting the product sales.

Average accuracy score, mean absolute error and max error for the Random Forest Regressor across the 10-fold stratified cross-validation is 87.72 percent, 3.15 and 0.44 error respectively which is quite impressive compared to other techniques. Simple linear Regressor produced the very poor results compared to the other techniques,

the average accuracy score, mean absolute error and max error across the 10-fold stratified cross-validation is 81.2 percent, 3.21, 0.49 error which is the poor one. And we can also observe from figure 4.13, Item price and outlet type grocery store are the critical features that will mainly influence the product sales. If the sales forecast is carried on every day across a large number of stores speed will play a key aspectin this process. Another useful metric to train the model, which will also play a crucial role while training several algorithms. The other important measure is the time required to train the model, which will also play a critical role while training different types of algorithms.

5.3 Contributions

As there are so many ongoing experiments that use statistical approaches and some traditional methods to focus on predicting item sales. Most researches have experimented by taking a single algorithm to predict sales. In this thesis Machine Learning algorithms such as Simple Linear Regression, K-Nearest Neighbor, Gradient Boosting algorithm, and Random Forest Regression are considered for prediction and the most effective metrics such as accuracy, mean absolute error, and max error are considered for measuring algorithm efficiency. This method will be very beneficial in the future for advanced item sales forecasting.

5.4 Validity Threats

5.4.1 Internal Validity

Proper preprocessing of data will be done. There may be a high chance of less accuracy if one can not perform data preprocessing properly.

5.4.2 External Validity

The data and findings used for the experiments are justified and relevant. This problem can be solved if the performance metrics and algorithms are not chosen properly. Proper selection of algorithms and performance metrics will be done.

Chapter 6

Conclusions and Future Work

6.1 Conclusion

Sales forecasting plays a vital role in the business sector in every field. With the help of the sales forecasts, sales revenue analysis will help to get the details needed to estimate both the revenue and the income. Different types of Machine Learning techniques such as K-Nearest Neighbor, Gradient Boosting Regression, Simple Linear Regression, and Random Forest Regression have been evaluated on food sales data to find the critical factors that influence sales to provide a solution for forecasting sales. After performing metrics such as accuracy, mean absolute error, and max error, the Random Forest Regression is found to be the appropriate algorithm according to the collected data and thus fulfilling the aim of this thesis.

6.2 Future Work

In future work one can attempt performance metrics such as time while predicting the sales. These metrics can play a crucial role in evaluating multiple Machine Learning algorithms. And also one can attempt to implement more accurate datain the continued study. Machine Learning has the advantage of analyzing data and key variables so that you can aim to develop a systematic approach using a varietyof Machine Learning techniques.

SOURCE CODE

Importing Modules

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
import statsmodels.api as sm
from sklearn.preprocessing import MinMaxScaler
import pickle
from os import path
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.neighbors import KNeighborsRegressor
from xgboost import XGBRegressor
from keras.models import Sequential
from keras.layers import Dense
from keras.wrappers.scikit_learn import KerasRegressor
```

Importing Dataset

```
data = pd.read_csv('train.csv')
stores = pd.read_csv('stores.csv')
features = pd.read_csv('features.csv')
data.head()
data.shape
data.tail()
stores.head()
stores.shape
stores.tail()
```

```
features.head()
features.shape
features.tail()
data.info()
stores.info()
features.info()
```

Handling missing values of features dataset

```
features["CPI"].fillna(features["CPI"].median(),inplace=True)
features["Unemployment"].fillna(features["Unemployment"].median(),inplace=True)
for i in range(1,6):
    features["MarkDown"+str(i)] = features["MarkDown"+str(i)]
    .apply(lambda x: 0 if x < 0 else x)
    features["MarkDown"+str(i)].fillna(value=0,inplace=True)
features.info()
```

Merging Training datasets and Merged Stores-Features Dataset

```
data = pd.merge(data,stores,on='Store',how='left')
data = pd.merge(data,features,on=['Store','Date'],how='left')
data['Date'] = pd.to_datetime(data['Date'])
data.sort_values(by=['Date'],inplace=True)
data.sort_values(by=['Date'],inplace=True)
data.set_index(data.Date, inplace=True)
data['IsHoliday_x'].isin(data['IsHoliday_y']).all()
data.drop(columns='IsHoliday_x',inplace=True)
data.rename(columns={"IsHoliday_y" : "IsHoliday"}, inplace=True)
data.info()
```

```
data.head()
```

Splitting Date Column

```
data['Year'] = data['Date'].dt.year
data['Month'] = data['Date'].dt.month
data['Week'] = data['Date'].dt.week
data.head()
```

Outlier Detection and Abnormalities

```
agg_data = data.groupby(['Store', 'Dept']).Weekly_Sales.agg(['max', 'min', 'mean', 'median', 'std']).reset_index()
agg_data.isnull().sum()
store_data = pd.merge(left=data,right=agg_data,on=['Store', 'Dept'],how ='left')
store_data.dropna(inplace=True)
data = store_data.copy()
del store_data
data['Date'] = pd.to_datetime(data['Date'])
data.sort_values(by=['Date'],inplace=True)
data.set_index(data.Date, inplace=True)
data.head()
data['Total_MarkDown'] = data['MarkDown1']+data['MarkDown2']+data['MarkDown3']+data['MarkDown4']+ data['MarkDown5']
data.drop(['MarkDown1','MarkDown2','MarkDown3','MarkDown4','MarkDown5'], axis = 1,inplace=True)
numeric_col = ['Weekly_Sales','Size','Temperature','Fuel_Price','CPI','Unemployment','Total_MarkDown']
data_numeric = data[numeric_col].copy()
```

```
data.shape
data = data[(np.abs(stats.zscore(data_numeric)) < 2.5).all(axis = 1)]
data.shape
y = data["Weekly_Sales"][data.Weekly_Sales < 0]
sns.displot(y,height=6,aspect=2)
plt.title("Negative Weekly Sales", fontsize=15)
plt.show()
data=data[data['Weekly_Sales']>=0]
data.shape
data['IsHoliday'] = data['IsHoliday'].astype('int')
data.to_csv('preprocessed_walmart_dataset.csv')
```

Data Visuallizations

Average Monthly Sales

```
plt.figure(figsize=(14,8))
sns.barplot(x='Month',y='Weekly_Sales',data=data)
plt.ylabel('Sales',fontsize=14)
plt.xlabel('Months',fontsize=14)
plt.title('Average Monthly Sales',fontsize=16)
plt.grid()
```

Monthly Sales of each Year

```
data_monthly = pd.crosstab(data["Year"], data["Month"],
values=data["Weekly_Sales"],aggfunc='sum')
data_monthly
fig, axes = plt.subplots(3,4,figsize=(16,8))
plt.suptitle('Monthly Sales for each Year', fontsize=18)
k=1
```

```
for i in range(3):
    for j in range(4):
      sns.lineplot(ax=axes[i,j],data=data_monthly[k])
      plt.subplots_adjust(wspace=0.4,hspace=0.32)
      plt.ylabel(k,fontsize=12)
      plt.xlabel('Years',fontsize=12)
      k+=1
plt.show()
```

Average Weekly Sales Storewise

```
plt.figure(figsize=(20,8))
sns.barplot(x='Store',y='Weekly_Sales',data=data)
plt.grid()
plt.title('Average Sales per Store', fontsize=18)
plt.ylabel('Sales', fontsize=16)
plt.xlabel('Store', fontsize=16)
plt.show()
plt.figure(figsize=(20,8))
sns.barplot(x='Store',y='Weekly_Sales',data=data)
plt.grid()
plt.title('Average Sales per Store', fontsize=18)
plt.ylabel('Sales', fontsize=16)
plt.xlabel('Store', fontsize=16)
plt.show()
```

Sales Vs Temperature

```
plt.figure(figsize=(10,8))
sns.distplot(data['Temperature'])
plt.title('Effect of Temperature',fontsize=15)
plt.xlabel('Temperature(in F)',fontsize=14)
plt.ylabel('Density',fontsize=14)
plt.show()
```

Holiday Distribution

```
plt.figure(figsize=(8,8))
plt.pie(data['IsHoliday'].value_counts(),labels=['No Holiday','Holiday'],autopct='%0.2f%%')
plt.title("Pie chart distribution",fontsize=14)
plt.legend()
plt.show()
```

Time Series Decompose

```
sm.tsa.seasonal_decompose(data['Weekly_Sales'].resample('MS')

.mean(), model='additive').plot()

plt.show()
```

One Hot Encoding

```
cat_col = ['Store','Dept','Type']

data_cat = data[cat_col].copy()

data_cat.tail()

data_cat = pd.get_dummies(data_cat,columns=cat_col)

data_cat.head()

data.shape

data = pd.concat([data,data_cat],axis =1)

data.shape

data.drop(columns=cat_col,inplace=True)

data.drop(columns=['Date'],inplace=True)
```

Data Normalization

```
num_col = ['Weekly_Sales','Size','Temperature','Fuel_Price','CPI','Unemployment',

'Total_MarkDown','max','min','mean','median','std']

minmax_scale = MinMaxScaler(feature_range=(0, 1))

def normalization(df,col):
```

```
        for i in col:
                arr = df[i]
                arr = np.array(arr)
                df[i] = minmax_scale.fit_transform(arr.reshape(len(arr),1))
         return df
data.head()
data = normalization(data.copy(),num_col)
data.head()
```

Corelation between features of datasets

```
plt.figure(figsize=(15,8))
corr = data[num_col].corr()
sns.heatmap(corr,vmax=1.0,annot=True)
plt.title('Correlation Matrix',fontsize=16)
plt.show()
```

Recursive Feature Elimination

```
feature_col = data.columns.difference(['Weekly_Sales'])
feature_col
radm_clf = RandomForestRegressor(oob_score=True,n_estimators=23)
radm_clf.fit(data[feature_col], data['Weekly_Sales'])
indices = np.argsort(radm_clf.feature_importances_)[::-1]
feature_rank = pd.DataFrame(columns = ['rank', 'feature', 'importance'])
for f in range(data[feature_col].shape[1]):
        feature_rank.loc[f] = [f+1,
        data[feature_col].columns[indices[f]],
         radm_clf.feature_importances_[indices[f]]]
```

```
feature_rank
x=feature_rank.loc[0:22,['feature']]
x=x['feature'].tolist()
print(x)
X = data[x]
Y = data['Weekly_Sales']
data = pd.concat([X,Y],axis=1)
data
```

Data Splitted into Training, Validation, Test

```
X = data.drop(['Weekly_Sales'],axis=1)
Y = data.Weekly_Sales
X_train,X_test,y_train,y_test = train_test_split(X,Y,test_size=0.20, random_state=50)
X
Y
```

Linear Regression Model

```
lr = LinearRegression(normalize=False)
lr.fit(X_train, y_train)
lr_acc = lr.score(X_test,y_test)*100
print("Linear Regressor Accuracy - ",lr_acc)
y_pred = lr.predict(X_test)
lr_df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
lr_df
plt.figure(figsize=(20,8))
plt.plot(lr.predict(X_test[:200]), label="prediction", linewidth=2.0,color='blue')
plt.plot(y_test[:200].values, label="real_values", linewidth=2.0,color='lightcoral')
```

```
plt.legend(loc="best")
plt.title('Prediction vs Real values(Linear Regression)',fontsize= 24)
plt.show()
```

Random Forest Regressor Model

```
rf = RandomForestRegressor()
rf.fit(X_train, y_train)
rf_acc = rf.score(X_test,y_test)*100
print("Random Forest Regressor Accuracy - ",rf_acc)
y_pred = rf.predict(X_test)
print("MAE" , metrics.mean_absolute_error(y_test, y_pred))
print("MSE" , metrics.mean_squared_error(y_test, y_pred))
print("RMSE" , np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
print("R2" , metrics.explained_variance_score(y_test, y_pred))
rf_df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
rf_df
plt.figure(figsize=(20,8))
plt.plot(rf.predict(X_test[:200]), label="prediction", linewidth=2.0,color='blue')
plt.plot(y_test[:200].values, label="real_values", linewidth=2.0,color='lightcoral')
plt.legend(loc="best")
plt.show()
```

K Neighbors Regressor Model

```
knn = KNeighborsRegressor(n_neighbors = 1,weights = 'uniform')
knn.fit(X_train,y_train)
knn = KNeighborsRegressor(n_neighbors = 1,weights = 'uniform')
knn.fit(X_train,y_train)
y_pred = knn.predict(X_test)
print("MAE" , metrics.mean_absolute_error(y_test, y_pred))
print("MSE" , metrics.mean_squared_error(y_test, y_pred))
print("RMSE" , np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
print("R2" , metrics.explained_variance_score(y_test, y_pred))
knn_df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
knn_df
plt.figure(figsize=(20,8))
plt.plot(knn.predict(X_test[:200]), label="prediction", linewidth=2.0,color='blue')
plt.plot(y_test[:200].values, label="real_values", linewidth=2.0,color='lightcoral')
plt.legend(loc="best")
plt.show()
```

XGboost Model

```
xgbr = XGBRegressor()
xgbr.fit(X_train, y_train)
xgb_acc = xgbr.score(X_test,y_test)*100
print("XGBoost Regressor Accuracy - ",xgb_acc)
y_pred = xgbr.predict(X_test)
xgb_df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
xgb_df
```

```
plt.figure(figsize=(20,8))
 plt.plot(xgbr.predict(X_test[:200]), label="prediction", linewidth=2.0,color='blue')
 plt.plot(y_test[:200].values, label="real_values", linewidth=2.0,color='lightcoral')
 plt.legend(loc="best")
 plt.show()
```

Comparing Models

```
acc = {'model':['lr_acc','rf_acc','knn_acc','xgb_acc'],
'accuracy':[lr_acc,rf_acc,knn_acc,xgb_acc]}
acc_df = pd.DataFrame(acc)
acc_df
plt.figure(figsize=(10,8))
sns.barplot(x='model',y='accuracy',data=acc_df)
 plt.show()
```

References

[1] Patrick Bajari, Denis Nekipelov, Stephen P Ryan, and Miaoyu Yang. Machinelearning methods for demand estimation. *American Economic Review*, 105(5):481–85, 2015.

[2] Kris Johnson Ferreira, Bin Hong Alex Lee, and David Simchi-Levi. Analytics foran online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management*, 18(1):69–88, 2016.

[3] Ankur Jain, Manghat Nitish Menon, and Saurabh Chandra. Sales forecasting forretail chains, 2015.

[4] Grigorios Tsoumakas. A survey of machine learning techniques for food sales prediction. *Artificial Intelligence Review*, 52(1):441–447, 2019.

[5] Xiaogang Su, Xin Yan, and Chih-Ling Tsai. Linear regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(3):275–294, 2012.

[6] Toby J Mitchell and John J Beauchamp. Bayesian variable selection in linear regression. *Journal of the american statistical association*, 83(404):1023–1032, 1988.

[7] Zheng Li, Xianfeng Ma, and Hongliang Xin. Feature engineering of machine-learning chemisorption models for catalyst design. *Catalysis today*, 280:232–238, 2017.

[8] Xinchuan Zeng and Tony R Martinez. Distribution-balanced stratified cross-validation for accuracy estimation. *Journal of Experimental & Theoretical Artificial Intelligence*,12(1):1–12, 2000.

[9] Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. On the stratification of multi-label data. In *Joint European Conference on Machine Learningand Knowledge Discovery in Databases*, pages 145–158. Springer, 2011.

[10] Chris Rygielski, Jyun-Cheng Wang, and David C Yen. Data mining techniques for customer relationship management. *Technology in society*, 24(4):483–502, 2002.

[11] Krzysztof J Cios, Witold Pedrycz, Roman W Swiniarski, and Lukasz Andrzej Kurgan. *Data mining: a knowledge discovery approach*. Springer Science & Business Media, 2007.

[12] Maike Krause-Traudes, Simon Scheider, Stefan Rüping, and Harald Meßner. Spatial data mining for retail sales forecasting. In *11th AGILE International Conference on Geographic Information Science*, pages 1–11, 2008.

[13] Stephen Marsland. *Machine learning: an algorithmic perspective*. CRC press, 2015.

[14] ML documentation https://www.mathworks.com/discovery/machine-learning.html). Accessed: 2020-04-22.

[15] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020.

Appendix A

Appendix

The graph below is a comparative analysis negative weekly sales.

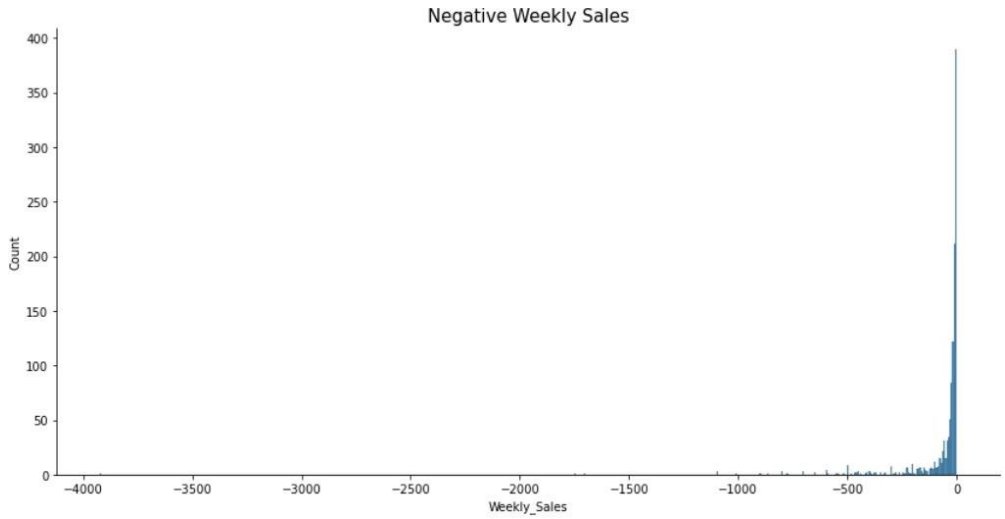

Figure A.1: Negative weekly sales

The graph below is a comparative analysis of monthly sales of each year.

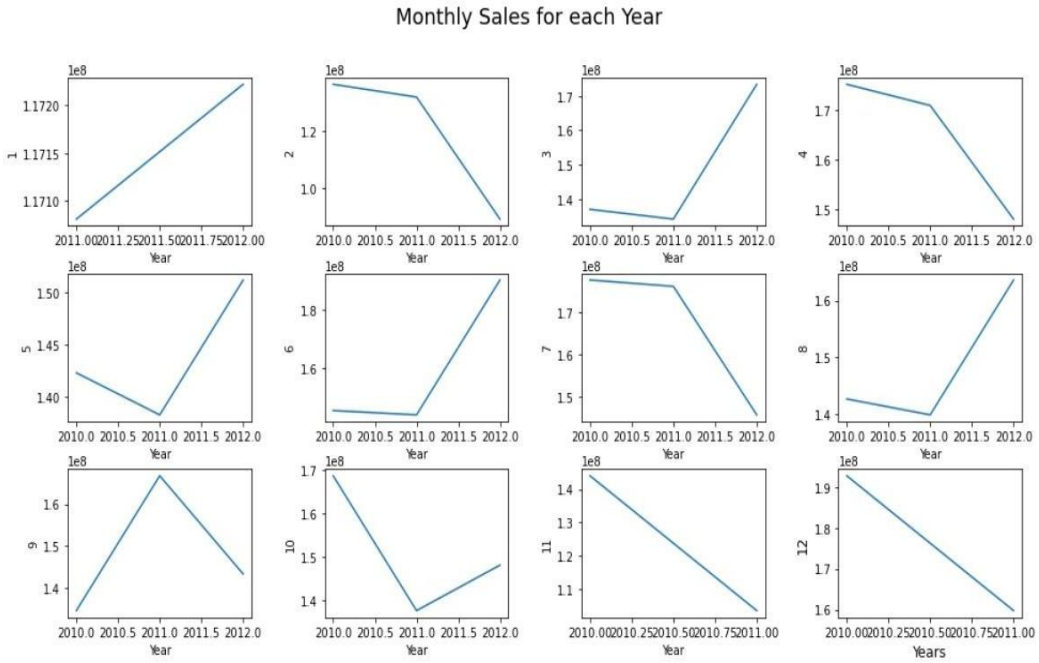

Figure A.2: Monthly sales of each year

The graph below is a comparative analysis of average monthly sales.

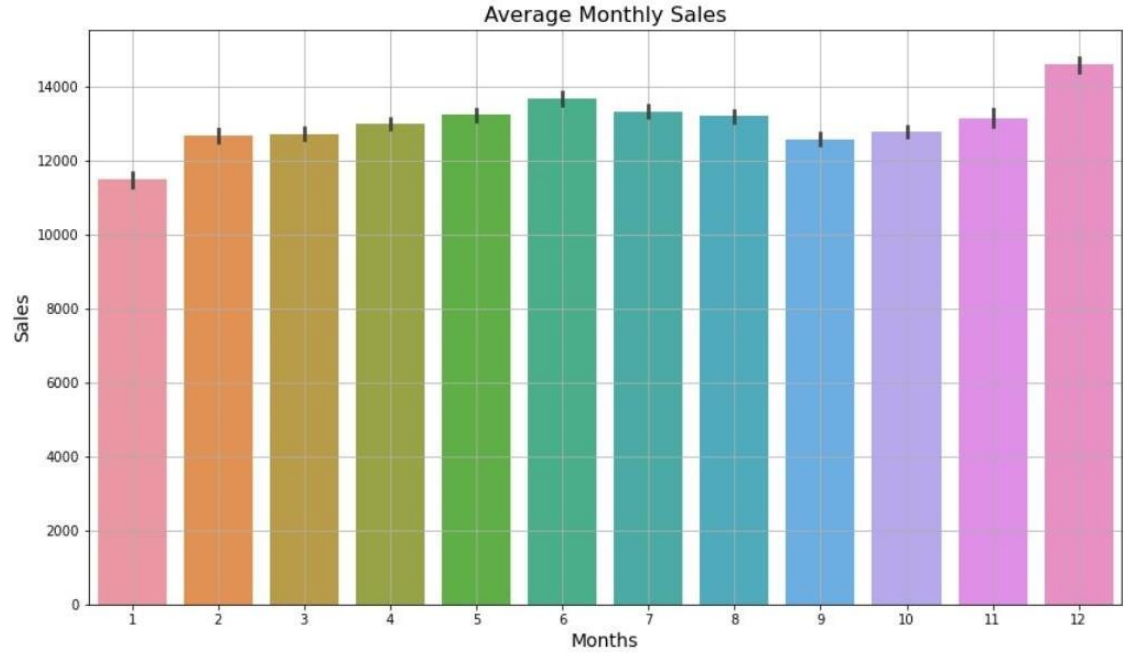

Figure A.3: Average monthly sales

The following graph illustrates the average sales of each store.

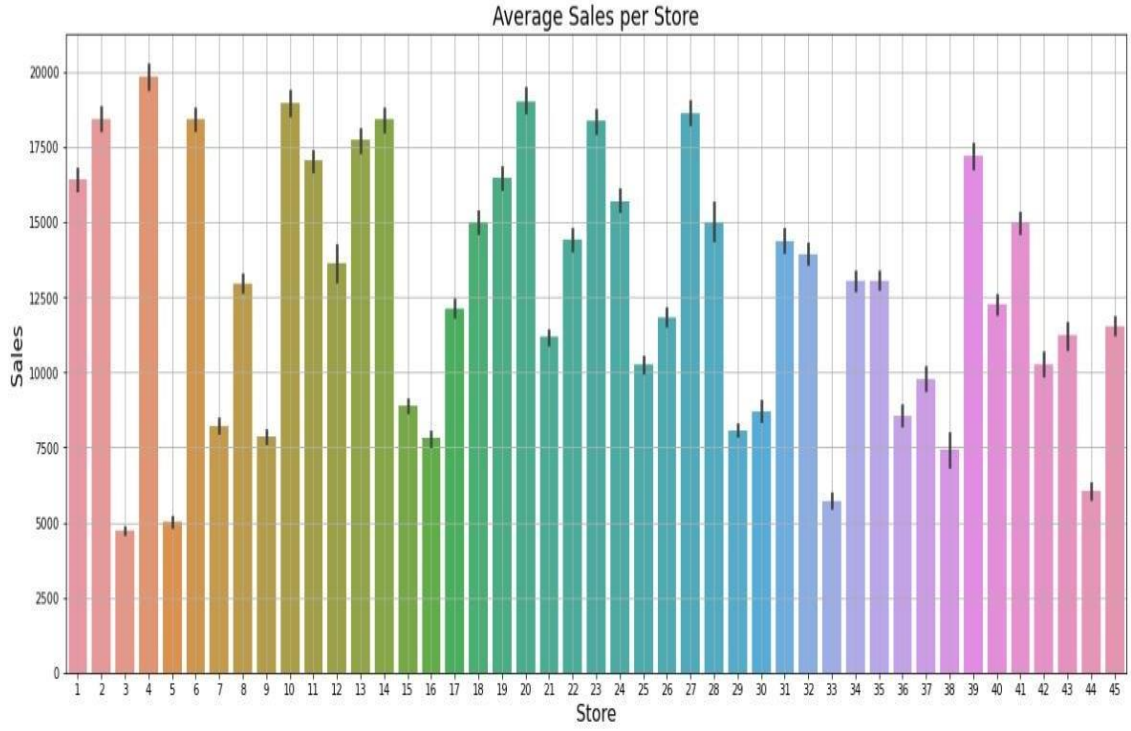

Figure A.4: Average sales of each store

The following graph illustrates the average sales per department.

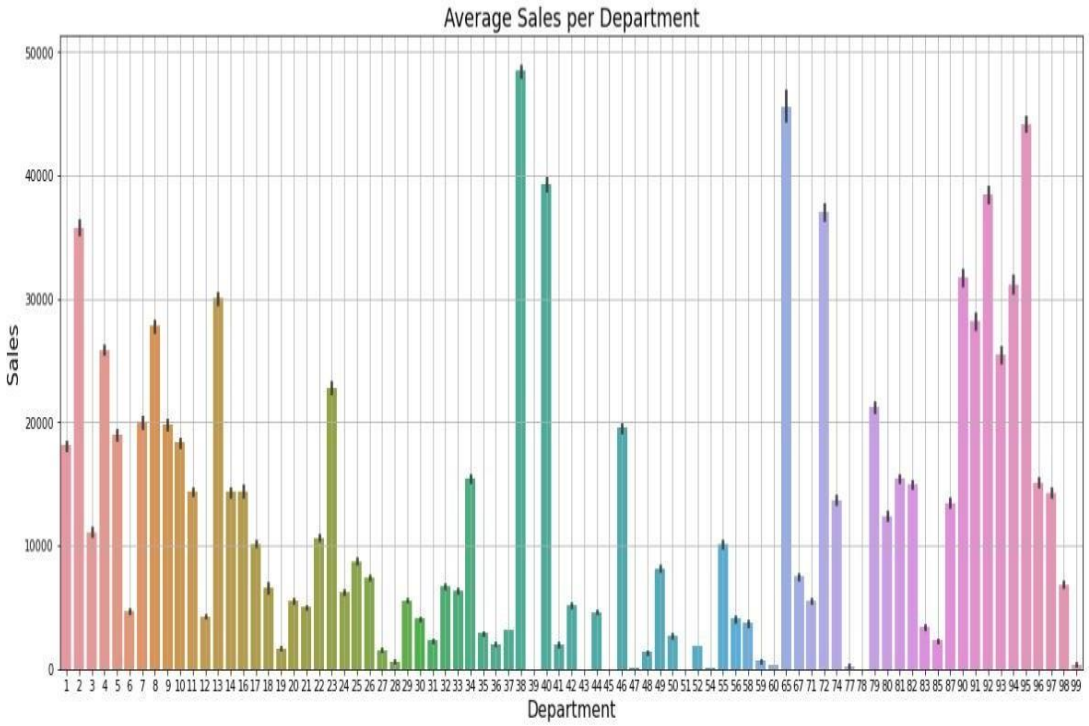

Figure A.5: Average sales per department