Project Report
On

Deploying Machine Learning Model on Cloud for Supermarket Sales Data Analytics and Forecasting

Submitted to

Sant Gadge Baba Amravati University

in Partial Fulfillment of the Requirement

for the Degree of

Bachelor of Engineering in

Computer Science and Engineering

Submitted by:

Mr. Sudhanshu Deshmukh

Mr. Sanket Deshmukh

Mr. Anshul Ghumadwar

Ms. Sakshi Deshmukh

Ms. Shruti Lambe

Under the Guidance of

Prof. C.M. Mankar

Department of Computer Science and Engineering

SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGINEERING,

SHEGAON – 444 203 (M.S.)

2022-23

SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGINEERING,
SHEGAON–444 203 (M.S.)
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that **Mr. Sudhanshu Deshmukh, Mr. Sanket Deshmukh, Mr. Anshul Ghumadwar, Ms. Sakshi Deshmukh** and **Ms. Shruti Lambe** students of final year B.E. in the year 2022-23 of Computer Science and Engineering Department of this institute has completed the project work entitled **"Deploying Machine Learning Model on Cloud for Supermarket Sales Data Analytics and Forecasting"** based on syllabus and has submitted a satisfactory account of her work in this report which is recommended for the partial fulfillment of degree of Bachelor of Engineering in Computer Science and Engineering.

Prof. C. M. Mankar
Project Guide

२६/5/23
Dr. S. B. Patil
Head of Department

Dr. S. B. Somani
Principal

SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGINEERING,

SHEGAON – 444 203 (M.S.)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project work entitled **“Deploying Machine Learning Model on Cloud for Supermarket Sales Data Analytics and Forecasting”** submitted by **Mr. Sudhanshu Deshmukh, Mr. Sanket Deshmukh, Mr. Anshul Ghumadwar, Ms. Sakshi Deshmukh** and **Ms. Shruti Lambe** students of final year B.E. in the year 2020-21 of Computer Science and Engineering Department of this institute, is a satisfactory account of his work based on syllabus which is recommended for the partial fulfillment of degree of Bachelor of Engineering in Computer Science and Engineering.

Internal Examiner

Date:

External Examiner

Date:

ABSTRACT

Currently, many supermarkets, shopping malls, and their various stores are operational in many cities and daily many more stores are getting added to the list leading to stiff competition. To beat the other competitors today supermarkets regularly audit the sales data of every product in the store to forecast future demands for that product and manage the total inventory properly. Proper optimization of inventory leads to greater profits and minimal losses. General trends and irregularities can be identified by data mining from the dataset. By analyzing the data, forecasting sales become easy with various machine-learning algorithms for supermarket chains. In this paper, we propose models using Linear regression, Ridge Regression, Random Forest Regression, and XGboost regressor for predicting sales of the Supermart and it was found that the trained model performs way better than other existing models in terms of accuracy. The trained model is then deployed over a cloud platform like AWS or Google Cloud which boosts the availability with a trouble-free user interface (UI).

ACKNOWLEDGEMENT

The real spirit of achieving a goal is through the way of excellence and lustrous discipline. I would have never succeeded in completing our task without the cooperation, encouragement and help provided to me by various personalities.

We would like to take this opportunity to express our heartfelt thanks to my guide ***Prof. C. M. Mankar****, for his esteemed guidance and encouragement, especially through difficult times. His suggestions broaden our vision and guided us to succeed in this work. We are also very grateful for his guidance and comments while studying part of our project and we learnt many things under his leadership.*

We extend our thanks to **Dr. S. B. Patil,** *Head of Computer Science & Engineering Department, Shri Sant Gajanan Maharaj College of Engineering, Shegaon for their valuable support that made us consistent performer.*

We also extend our thanks to **Dr. S. B. Somani***, Principal, Shri Sant Gajanan Maharaj College of Engineering, Shegaon for his valuable support.*

Also, we would like to thanks to all teaching and non-teaching staff of the department for their encouragement, cooperation and help. Our greatest thanks are to all who wished us success especially our parents, our friends whose support and care makes us stay on earth.

1. **Mr. Sudhanshu Deshmukh**
2. **Mr. Sanket Deshmukh**
3. **Mr. Anshul Ghumadwar**
4. **Ms. Sakshi Deshmukh**
5. **Ms. Shruti Lambe**

Session 2022-23

CONTENTS

List of Figures and Tables

List of Snapshots

Abbreviations

ML	Machine Learning
AI	Artificial Intelligence
GNMT	Google Neural Machine Translation
KDD	Knowledge Discovery in Databases
SVM	Support Vector Machine
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
MSE	Mean Square Error
PCA	Principal Component Analysis
RF	Random Forest
GFS	Genetic Fuzzy Systems
MSE	Mean Square Error
KNN	K-Nearest Neighbor

CHAPTER 1
INTRODUCTION

INTRODUCTION

1.1 PREFACE

Sales prediction is used for the prediction of future sales based on data of the past years of the supermarket chain. Sales forecasting can help to prevent a shortage of most selling products and prevents the build-up of stock of less-selling product. Ultimately it leads to higher profits. Sales forecasting has a crucial role in proper inventory management and prediction of demands during holidays or festivals. Various trends and patterns are observed to increase sales and anomalies are ruled out. Supermarket chains do the forecast to limit the use of marketing spent and to manage the supply chain properly to supply [1]. Data mining techniques are potent in tuning a large amount of data and are crucial for sales prediction and sales prediction and are necessary for budget planning [2]. Population demographics around the store also affect sales, the capacity of the store and other components should be evaluated. For Deployment purposes, a cloud platform is more suitable than traditional deployment methods [5]. Marketing strategies can be made by understanding the sales pattern. Let’s say any particular product is being sold in higher quantities so it helps to derive crucial insights that is why the product is being sold most and helps companies to take marketing decisions.

Every item in its shopping centers and Supermarts is tracked in order to forecast future customer demand and enhance inventory management. Big Mart is a massive network of stores that spans the globe. Big Mart's trends are Data scientists analyze those tendencies per product and store in order to generate potential centers, which is highly relevant. Using a machine to forecast Big Mart transactions allows data scientists to try multiple patterns by shop and product to attain the best results. Many businesses rely largely on their knowledge base and require forecasting of market tendencies.

Population statistics around the store also affect sales, and the capacity of the store and many more things should be considered. Because every business has strong demand, sales forecasts play a significant part in a retail center. A stronger prediction is always helpful in developing and enhancing corporate market strategies, which also help to increase awareness.

1.2 MOTIVATION

Inventories can either make or break a business. Having an inventory management solution is a must in today’s modern digital era. Our solution focuses on solving this problem of mismanaged inventories for several businesses by providing them with an interactive and functional dashboard powered by Machine Learning and Data Science techniques to attain maximum efficiency. Our project forecasts the most optimal quantity to possess in an inventory for a particular month whilst taking into account several factors such as the history of product sales, seasonal trends like what the consumers demand at a particular period of time, external factors which affect the sales, and financial insights. Our solution also puts light on giving comprehensive details regarding a particular business by the virtue of in-depth analysis and interactive visualizations. These models can be implemented in various areas and taught to fit management expectations, allowing for precise procedures to be taken to attain the organization's goal. In this research, the instance of Supermart, a one-stop shopping center, has been discussed in order to anticipate the sales of various types of things and to analyze the effects of various factors on the sales of the items. Using various components of a dataset acquired for Supermart and the methods used to develop a predictive model, high-accuracy findings are obtained, and these observations can be used to make sales-improving decisions. Sales forecasting is critical in the development of a business. It is a critical component of business intelligence. Sales forecasting and prediction provide insight into how a firm should manage its workforce, including labour, cash flow, and resources. It is a method for predicting future performance based on past and current sales data. Estimating future sales is an important aspect of every company's financial planning. It enables businesses to forecast both short- and long-term performance. Accurate sales predictions help businesses make informed decisions, which leads to better supply chain management, more earnings, and improved customer experiences. It is an essential component of beginning a new firm because it aids in the efficient management of existing resources. Furthermore, understanding customer behavior gets easier with the use of data and learned insights.

1.3 PROBLEM STATEMENT

The data scientists at Big Mart have collected 2013 sales data for 1559 products across 10 stores in different cities. Also, certain attributes of each product and store have been defined. The aim is to build a predictive model and find out the sales of each product at a particular store. Using this model, Big Mart will try to understand the properties of products and stores which play a key role in increasing sales.

1.4 OBJECTIVE

- The goal of this framework is to predict future sales based on previous year's data using MLTechniques.
- To increase revenue, build a solid sales trend forecast system that is implemented using ML technologies.
- As a result, analyzing important appearance includes determining the most effective impact upon commodity demand.
- That resolve the ML invention is best for demand predicting.
- Choosing various verification to match the efficiency of the ML algorithms in use.

1.5 SCOPE OF PROJECT

1. This project will help to analyze critical features that will most influence sales of the product.
2. It will Convert data into an appropriate form using various preprocessing techniques for the implementation of Machine Learning algorithms.
3. Observing and analyzing forecasting of sales.
4. Predict and analyze prediction of sales.

1.6 ORGANIZATION OF PROJECT

Chapter 1: It gives an Introduction of the project.

Chapter 2: Literature Survey of the research papers referred to get an idea of the previous work done on this project.

Chapter 3: After reviewing, the methodology of how the project can be executed.

Chapter 4: The implementation details that we have followed to complete the project.

Chapter 5: How the project was deployed on frontend using Flask frontend framework.

Chapter 6: The conclusion derived from this project.

Chapter 7: Details of the research papers referred.

1.7 REQUIREMENT SPECIFICATION

- **SOFTWARE REQUIREMENTS:**
    - Jupyter notebook
    - Vs code editor
    - Winscp
    - Inspect tool

- **HARDWARE REQUIREMENTS:**
    - Minimum 8GB RAM
    - Minimum Intel i5 processor
    - Keyboard, Mouse, Screen
    - Minimum 80GB HDD

CHAPTER 2
LITERATURE SURVEY

LITERATURE SURVEY

2.1 PREVIOUS WORK

Sales predictions reveal how a company should manage its staff, cash flow, and resources. This is a necessary precondition for enterprise planning and decision-making. It enables businesses to efficiently construct their business plans. Learning algorithms used in classification and model categories, such as linear Regression, Ridge Regression, Random Forest, Decision Tree, and XG Boost, are appropriate for sales forecasting. The regression technique is used to forecast, model time series, and determine the cause-effect relationship between variables. A handful of those who have performed prediction on various sales data are given below.

B. Kumar Jha et al.,[4] research focuses on ARIMA and FB Prophet Additive models for the prediction of sales. Sales data for furniture stores were examined and predictions made using FB Prophet Additive resulted to be very near too actual. It shows good accuracy for time series data.

Sai Nikhil et al.,[20] research focuses on developing a machine learning model that can anticipate product sales across several venues. It has described the project's setting, data correlation, and data preparation in detail. It provides a quick categorization as well as an introduction to machine learning techniques. This research compared multiple ML methods, including simple linear regression, gradient boosting regression, support vector regression, and random forest regression, to find the best fit. The results are given as absolute mean & maximum errors. It has been demonstrated that random forest method is more accurate than other algorithms.

Y. Sener et al.,[5] studied a case in which an ML model was deployed with Amazon Web Services (AWS) and its tools. The study also states the importance of cloud platforms for deployment over on-premise deployment. Other pros of cloud deployment were studied for new organizations and startups.

G. Behera et al.,[6] For the purpose of forecasting future sales for Bigmart, research focuses on hyperparameter tuning (HPT) using grid search optimization. For training and testing datasets, this method produced findings that were more accurate and had lower RMSE and MAE.

S. Cheriyan et al.,[21] The data tuning process and several data mining approaches were explored in this study. The sales generated in an e-fashion store over three years are the dataset used in this research article. There are up to 85000 results in the comparison. In this work, linear regression, decision trees, and gradient boosted trees were compared. Gradient enhanced trees, according to the research, are more accurate.

A. Krishna et al.,[22] In this study, regular regression approaches and boosting algorithms were thoroughly contrasted. The collection has 8523 samples and 12 characteristics. Regular algorithms include linear regression, polynomial regression, lasso regression, and ridge regression, whilst boosting techniques include AdaBoost and Gradient Boost. These algorithms were developed using Python 3.6 and Sklearn 0.19.1. The algorithm with the lowest RMSE value outperformed the one with the highest. AdaBoost, a boosting approach, has a lower RMSE value than Gradient Boost. According to this article, Gradient Boost outperformed the AdaBoost approach.

R. P and S. M [23] research work gave a simple and concise description of algorithm accuracies. Screening datasets from the Kaggle website with a sample size of 5000 train datasets and 8000 test datasets were used in this study. An architectural diagram shows why the model is being offered in the methodology section. They calculated accuracy using several regression techniques. The ridge regression and the XG-Boost regression were shown to be more accurate.

Yua et al. [24] forecasted magazine and newspaper sales using Support Vector Regression. Support Vector Regression was utilized because it solved the over-fitting problem while simultaneously achieving the lowest structural risk rather than the lowest empirical risk. E. Hadavandi et al.,[25] employed a combination of Genetic Fuzzy Systems (GFS) and data clustering to anticipate printed circuit board sales. They used K-means clustering to build K clusters of all the data records. The clusters were then fed into independent Genetic Fuzzy Systems (GFS) capable of database tweaking and rule-based extraction.

Sr. no.	Title	Name of Journal/ conference	Author name	Key Findings	Approach	Conclusion
[1]	Predictive Analysis for Big Mart Sales Using Machine Learning Algorithms	ICICCS,2021	R.P and S.M	It was found that the general linear model using the principal component analysis and the random forest techniques produce better results which are been decided by the RMSE values.	Linear Regression, Ridge Regression, Random Forest, Decision Tree, XGBoost these algorithms.	Ridge regression and XGboost regression give better prediction accuracy than linear and polynomial regression approaches.
[2]	Intelligent Sales Prediction Using Machine Learning Techniques	iCCECE, 2020	Sunitha Cheriyan, Shaniba Ibrahim, Saju Mohanan, Susan Treesa.	Various types of mining approaches and data-tuning processes to forecast sales as it helps the model become comprehensive and reliable. In this work, the dataset for the e-fashion store was studied and various regression algorithms were compared.	Various prediction methods, sales forecasting strategies and Expectation Maximization (EM) algorithm	Based on the performance, it is understood that Gradient Boost Algorithm is showing 98% overall accuracy and the second stands Decision Tree Algorithms with nearly 71% overall accuracy and followed by Generalized Linear Model with 64% accuracy.
[3]	Sales-forecasting of Retail Stores using Machine Learning Techniques	IEEE, 2018	Akshay Krishna, Akhilesh V, Animikh Aich, Chetana Hegde.	The performance of the individual model was comparatively lower than the hybrid model.	Multiple Regression, Polynomial Regression, Ridge Regression, Lasso Regression etc. along with various boosting algorithm like AdaBoost, Gradient Tree Boosting so as to get the maximum accuracy.	The literature in this field shows that not much work has been done in swarm intelligence technique in effectively training the prediction models. The Genetic Algorithm (GA) is a potential candidate for training the ANN models.

CHAPTER 3
METHODOLOGY

3.1 MACHINE LEARNING

Machine learning is a subset of artificial intelligence (AI) that involves the use of algorithms and statistical models to enable computers to learn from and make predictions or decisions based on data without being explicitly programmed. Machine learning algorithms are designed to recognize patterns, make predictions, classify data, and optimize performance based on feedback from data inputs.

The general process of machine learning involves several key steps:

Data Collection: Machine learning requires large amounts of data for training and validation. Data can be collected from various sources, such as databases, sensors, or online platforms.

Data Preprocessing: Once the data is collected, it needs to be cleaned, normalized, and transformed into a format suitable for machine learning algorithms. This step may also involve feature engineering, which is the process of selecting relevant features from the data to improve model performance.

Model Selection: There are various types of machine learning models, such as supervised learning, unsupervised learning, reinforcement learning, and deep learning. The appropriate model needs to be selected based on the nature of the data and the specific problem being solved.

Model Training: In this step, the selected machine learning model is trained using the preprocessed data. The model learns from the data and tries to identify patterns or relationships that can be used for making predictions or decisions.

Model Evaluation: Once the model is trained, it needs to be evaluated for its performance on a separate set of data that was not used for training. Common evaluation metrics include accuracy, precision, recall, F1 score, and area under the curve (AUC).

Model Optimization: Based on the evaluation results, the model may need to be fine-tuned or optimized to improve its performance. This may involve adjusting hyperparameters, feature selection, or using ensemble methods to combine multiple models for better results.

Model Deployment: After the model is optimized, it can be deployed in a real-world environment for making predictions or decisions. This may involve integrating the model into a production system, creating an API, or developing a user-friendly interface for end-users.

Model Monitoring and Maintenance: Machine learning models need to be monitored and maintained to ensure their continued accuracy and performance. This may involve updating the model with new data, retraining the model periodically, and addressing any issues or biases that may arise during deployment.

Machine learning is used in a wide range of applications, including but not limited to image and speech recognition, natural language processing, recommendation systems, fraud detection, healthcare, finance, and autonomous vehicles. It continues to be an exciting field with ongoing advancements and innovations that have the potential to transform many aspects of our lives.

3.2 TYPES OF MACHINE LEARNING TECHNIQUES

There are four types of machine learning: supervised, unsupervised, semi-supervised, and reinforcement learning. A supervised learning model must accomplish two fundamental tasks: classification and regression. Classification is concerned with predicting a nominal class label, whereas regression is concerned with predicting a numerical value for the class label. Building a regression model mathematically is all about identifying the relationship between the class label and the input predictors. Attributes are another name for predictors. In statistics, the predictors are referred to as independent variables, whereas the class label is referred to as a dependent variable [14]. A regression model depicts the relationship between dependent and independent variables. Any new data is plugged into the relationship curve to find the prediction once this is learned during the training phase. This simplifies the machine learning problem to a mathematical equation to solve [8].

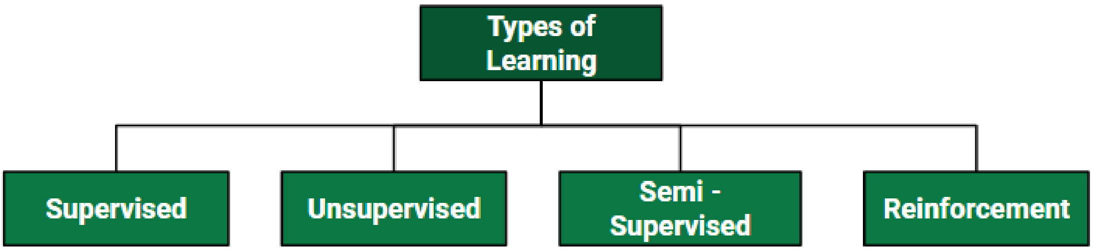


Figure 3.1: Types of Machine Learning

3.2.1 Supervised Learning:

Supervised learning is a type of machine learning where a model is trained using labeled data, where the input data is paired with corresponding output labels. The model learns to make predictions or decisions based on this labeled data. In supervised learning, the goal is to minimize the difference between the model's predicted output and the actual output, which is known as the "label" or "target" value. Supervised learning can be further classified into two main categories:

Classification: In classification, the model learns to predict discrete categories or classes for new input data. For example, a classification model can be trained to predict whether an email is spam or not spam, or whether a customer will churn or not churn in a subscription service. The output of a classification model is typically a probability or a class label indicating the predicted category.

Regression: In regression, the model learns to predict continuous values for new input data. For example, a regression model can be trained to predict the price of a house based on its features such as location, size, and number of rooms. The output of a regression model is a continuous value that represents the predicted output, such as a numeric value or a range of values.

Supervised learning is widely used in various applications, such as image and speech recognition, natural language processing, fraud detection, medical diagnosis, and many others. It relies on labeled data to train the model, and the model's performance is evaluated based on its ability to accurately predict the correct output based on new, unseen data. In supervised learning, the ML algorithm is given a small training dataset to work with. This training dataset is a smaller part of the bigger dataset and serves to give the algorithm a basic idea of the problem, solution, and data points to be dealt with.

The training datasetis also very similar to the final dataset in its characteristics and provides the algorithm with the labeled parameters required for the problem [9]. Supervised learning is the one where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output.

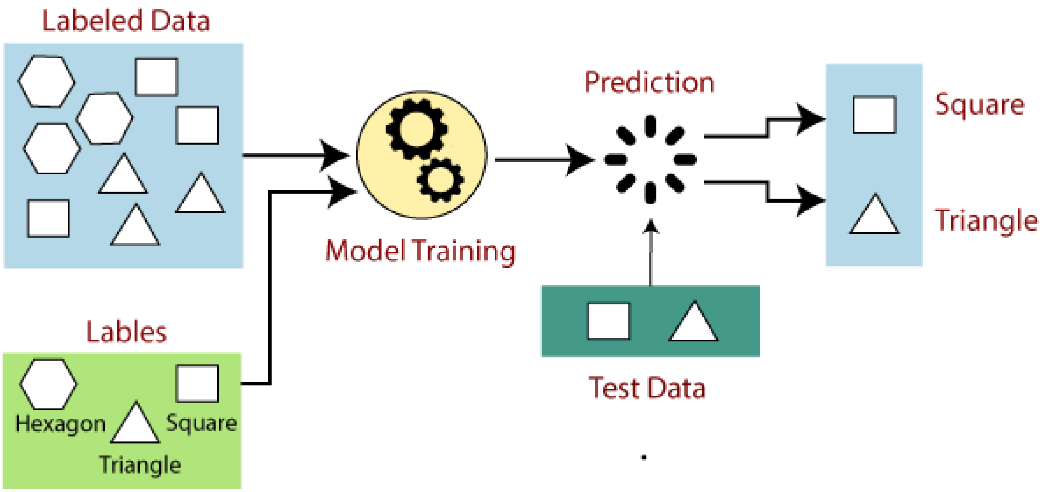


Figure 3.2: Supervised Learning

3.2.2 Unsupervised Learning:

Unsupervised learning is a type of machine learning where an algorithm learns from data without any labeled examples or explicit guidance from a human supervisor. In unsupervised learning, the algorithm is tasked with discovering patterns, structures, or relationships in the data on its own, without being provided with predefined labels or categories to predict.

Unsupervised learning algorithms explore the data and identify inherent patterns or structures without any prior knowledge. Some common techniques used in unsupervised learning include clustering, dimensionality reduction, and anomaly detection.

Clustering is a common unsupervised learning technique where the algorithm groups similar data points together based on their similarities or distances in a multi-dimensional space. Clustering algorithms can be used for tasks such as customer segmentation, image segmentation, or document grouping.

Dimensionality reduction is another unsupervised learning technique that aims to reduce the number of features or variables in the data while retaining important information. This can be useful for tasks such as visualizing high-dimensional data or reducing computational complexity.

Anomaly detection is a type of unsupervised learning that focuses on identifying data points that deviate significantly from the expected or normal behavior. Anomaly detection algorithms are used in various applications such as fraud detection, intrusion detection, or equipment failure prediction.

Unsupervised learning has several advantages, such as its ability to uncover hidden patterns or structures in data without labeled examples, its applicability to a wide range of domains, and its potential for discovering new insights or anomalies in the data. However, it also has challenges, such as the lack of labeled data for model evaluation, the potential for identifying spurious patterns, and the need for human interpretation of the results.

In supervised learning, the goal is to learn mapping from the input to an output whose correct values are provided by a supervisor. But, in unsupervised learning, the goal is to find the regularities in the input such that certain patterns occur more often than others and to learn to see what generally happens and what does not. Examples on speech recognition, document clustering, and image compression go well with unsupervised learning.

The unsupervised model learns through observation and finds structures in the data. Once the model is given a dataset, it automatically finds patterns and relationships in the dataset by creating clusters in it. Unsupervised learning is used for raw datasets. Its main task is to convert raw data to structured data. Unsupervised learning is where we only have input data (X) and no corresponding output variables i.e., Y

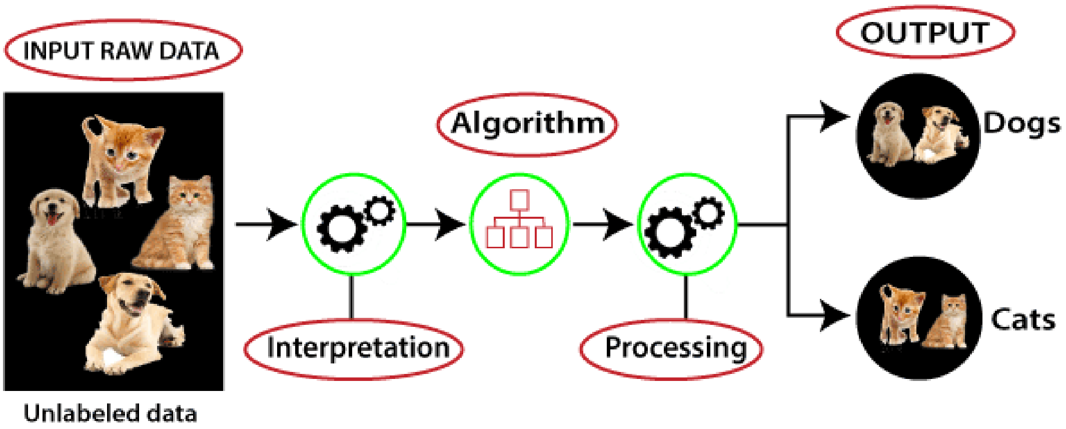


Figure 3.3: Unsupervised Learning

3.3 METHODOLOGY

This thesis is divided into four sections: a literature review, a data analysis and comprehension phase, an implementation phase, and an evaluation phase. Except for the implementation phase, which had several iterative sub-phases, these phases were typically conducted sequentially.

The major purpose of the literature study section was to read about and become familiar with the following topics: decision trees, random forest, extreme gradient boosting, support vector machine for regression, ensemble models, and hyper-parameter tuning. Techniques for feature selection were also examined because data complexity and a large number of features can be an impediment to performance improvement. Data analysis is typically performed as the initial step in any machine learning work flow, and in this experiment, data analysis is performed as the first stage and is detailed later in this chapter. This step entails delving deeply into the data in order to fully comprehend it and extract some correlations between variables.

The implementation phase consisted of the following parts:

1. Applying necessary feature engineering steps in order to prepare the data for the next step of model development.

2. Developing models using the complete set of features.

3. Performing hyper-parameter optimization and re-train the models as in the second step.

4. Building ensembles of the developed models.

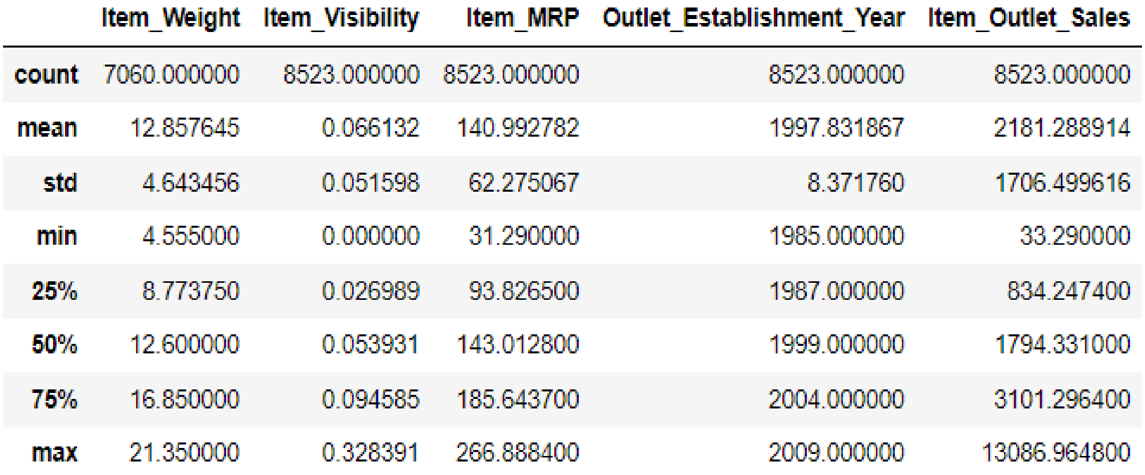
5. Choosing a subset of features according to a defined procedure (explained in the next chapter) and develop new models using only the subset of features.

In this stage we have done the following three steps:

1. Data overview
2. Feature Selection
3. Data Correlation Methods

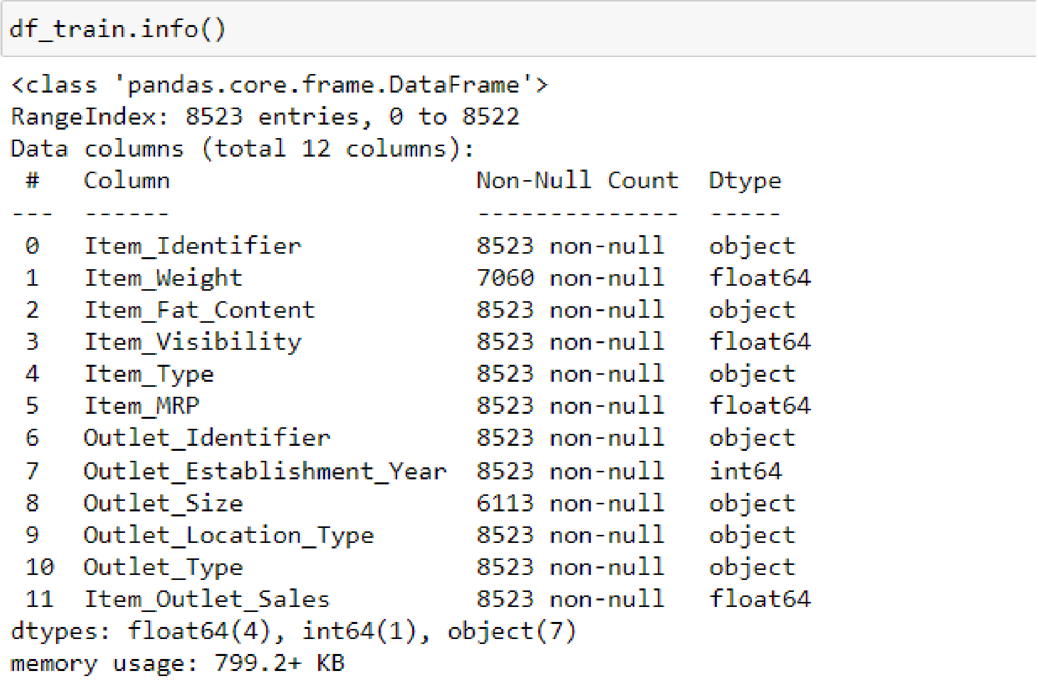
3.3.1 Data overview:

This thesis contains labelled sales data from various items from various outlets that provide information like as item kind, item price, outlet type, and so on. These data were gathered from a variety of sources and will be utilized to train and develop the Machine Learning model. There are 8523 occurrences and 12 attributes in the dataset under consideration. The dataset has been appropriately separated into training and testing data, as described in the sections below.



	Item_Weight	Item_Visibility	Item_MRP	Outlet_Establishment_Year	Item_Outlet_Sales
count	7060.000000	8523.000000	8523.000000	8523.000000	8523.000000
mean	12.857645	0.066132	140.992782	1997.831867	2181.288914
std	4.643456	0.051598	62.275067	8.371760	1706.499616
min	4.555000	0.000000	31.290000	1985.000000	33.290000
25%	8.773750	0.026989	93.826500	1987.000000	834.247400
50%	12.600000	0.053931	143.012800	1999.000000	1794.331000
75%	16.850000	0.094585	185.643700	2004.000000	3101.296400
max	21.350000	0.328391	266.888400	2009.000000	13086.964800

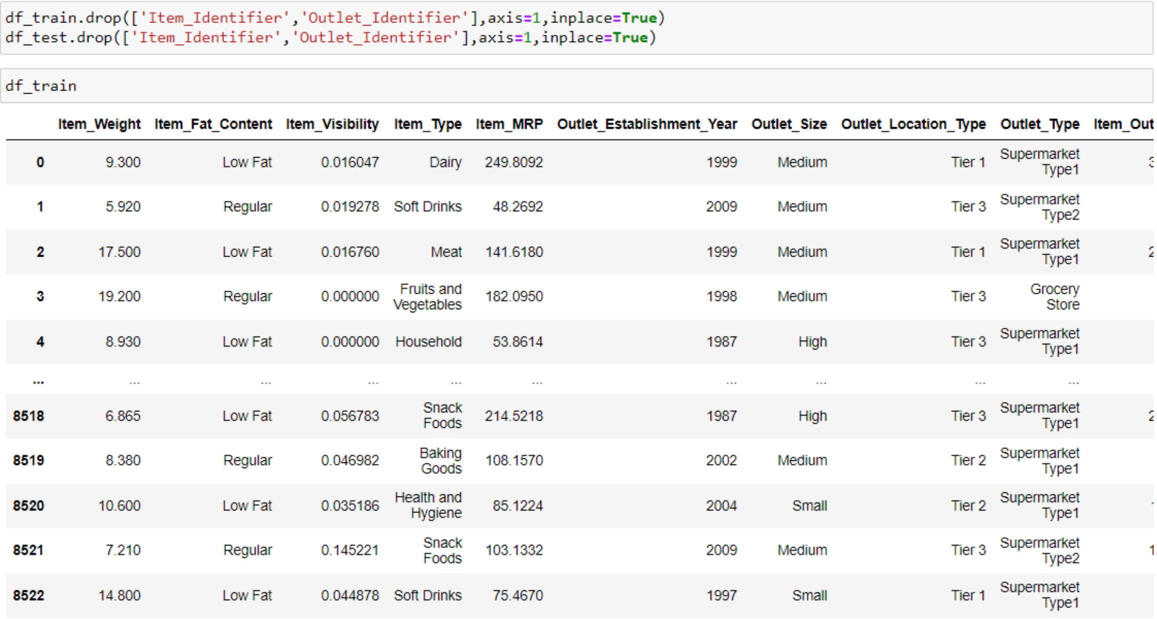
Figure 3.4: Data Overview



```
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 12 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Item_Identifier            8523 non-null   object 
 1   Item_Weight                7060 non-null   float64
 2   Item_Fat_Content           8523 non-null   object 
 3   Item_Visibility            8523 non-null   float64
 4   Item_Type                  8523 non-null   object 
 5   Item_MRP                   8523 non-null   float64
 6   Outlet_Identifier          8523 non-null   object 
 7   Outlet_Establishment_Year  8523 non-null   int64  
 8   Outlet_Size                6113 non-null   object 
 9   Outlet_Location_Type       8523 non-null   object 
 10  Outlet_Type                8523 non-null   object 
 11  Item_Outlet_Sales          8523 non-null   float64
dtypes: float64(4), int64(1), object(7)
memory usage: 799.2+ KB
```

Figure 3.5: Dataset Summary



```
df_train.drop(['Item_Identifier','Outlet_Identifier'],axis=1,inplace=True)
df_test.drop(['Item_Identifier','Outlet_Identifier'],axis=1,inplace=True)
```

```
df_train
```

	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Type	Outlet_Type	Item_Out
0	9.300	Low Fat	0.016047	Dairy	249.8092	1999	Medium	Tier 1	Supermarket Type1	3
1	5.920	Regular	0.019278	Soft Drinks	48.2692	2009	Medium	Tier 3	Supermarket Type2	
2	17.500	Low Fat	0.016760	Meat	141.6180	1999	Medium	Tier 1	Supermarket Type1	2
3	19.200	Regular	0.000000	Fruits and Vegetables	182.0950	1998	Medium	Tier 3	Grocery Store	
4	8.930	Low Fat	0.000000	Household	53.8614	1987	High	Tier 3	Supermarket Type1	
...	...	...	...	...	...	...	...	...	...	
8518	6.865	Low Fat	0.056783	Snack Foods	214.5218	1987	High	Tier 3	Supermarket Type1	2
8519	8.380	Regular	0.046982	Baking Goods	108.1570	2002	Medium	Tier 2	Supermarket Type1	
8520	10.600	Low Fat	0.035186	Health and Hygiene	85.1224	2004	Small	Tier 2	Supermarket Type1	
8521	7.210	Regular	0.145221	Snack Foods	103.1332	2009	Medium	Tier 3	Supermarket Type2	1
8522	14.800	Low Fat	0.044878	Soft Drinks	75.4670	1997	Small	Tier 1	Supermarket Type1	

Figure 3.6: Selecting features on general requirements

3.3.2 Feature Selection:

There are numerous aspects that can improve the effectiveness of a Machine Learning model on any given task. Data correlation is one approach of feature selection that will have a significant impact on the model's performance. This will relieve a significant amount of burden on the Machine Learning model during preprocessing and data purification. The data variables used to train the Machine Learning model would have a significant impact on the model's efficiency. The model output will be lowered as a result of the supplied irrelevant features. The feature selection approach is an efficient way to remove data redundancy and irrelevant data, which reduces computing time, improves accuracy, and improves model understanding.

Feature selection is an important step in the machine learning pipeline as it can lead to improved model performance, reduced complexity, and faster training and inference times. By selecting a subset of relevant features, feature selection can help to improve the interpretability and generalization of machine learning models, and can also reduce the risk of overfitting, which occurs when a model learns to perform well on the training data but does not generalize well to unseen data.

3.3.3 Data Correlation Method:

Data correlation is a statistical measure that quantifies the strength and direction of the linear relationship between two or more variables in a dataset. Correlation can be used as a feature selection method to identify relevant features that are highly correlated with the target variable or with other features in the dataset.

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 relationbetween any two attributes then it is said to be no correlation.

If there is 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 [15].

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

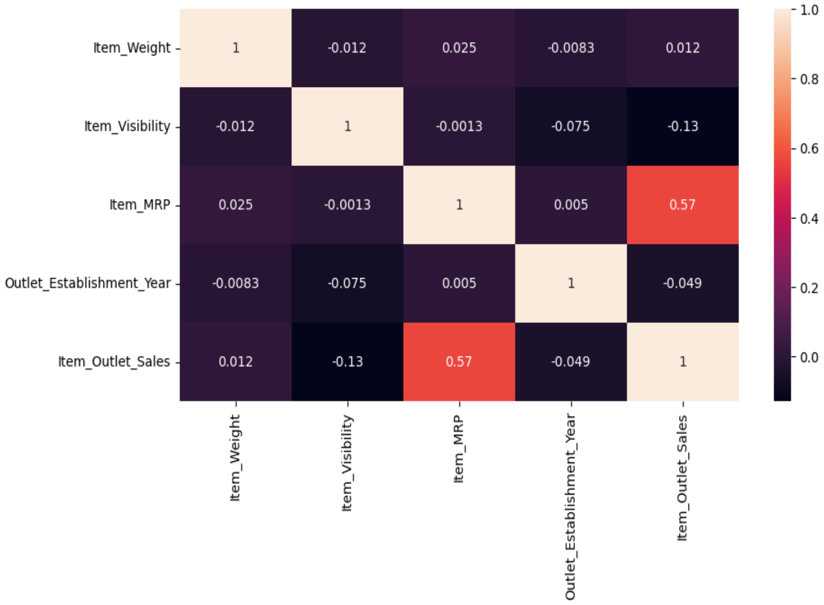


Figure 3.7: Heat Map

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 attributes is shown in figure 3.4. To improve the efficiency of the Machine Learning model, the attributes that have negative correlations were removed [20]. It is a statistic measuring the 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.

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 related to it more accurate. The more accurate the prediction, the stronger the relationship between the variables.

3.4 MACHINE LEARNING ALGORITHMS

We have also used various algorithms and processes from following in our project to train the model and they are:

- Simple Linear Regression
- Random Forest Regression
- XGBoost Regression
- Ridge Regression

3.4.1 Simple Linear Regression:

Simple linear regression is a statistical method used to model the relationship between two variables, where one variable is considered as the dependent variable (also known as the response or outcome variable) and the other variable is considered as the independent variable (also known as the predictor or explanatory variable). It assumes a linear relationship between the two variables, which means that the relationship can be represented by a straight line.

The main goal of simple linear regression is to determine the best-fitting line through the data points that minimizes the residuals (the differences between the observed values and the predicted values). The equation of the simple linear regression model can be represented as:

$Y = \beta 0 + \beta 1 * X + \varepsilon$

Where:
Y represents the dependent variable (response variable)
X represents the independent variable (predictor variable)
$\beta 0$ represents the intercept (the value of Y when X = 0)
$\beta 1$ represents the slope (the change in Y for a unit change in X)
ε represents the error term (the random variation not explained by the model)

The parameters β0 and β1 are estimated using various statistical techniques, such as the least squares method, which aims to minimize the sum of squared residuals. Once the estimates of β0 and β1 are obtained, they can be used to make predictions of the dependent variable (Y) for a given value of the independent variable (X) using the equation of the regression line.

Simple linear regression is commonly used in various fields, such as economics, social sciences, finance, and marketing, to analyze and predict the relationship between two variables and understand how changes in the predictor variable affect the response variable. It is a simple and widely used method for exploring and modeling linear relationships between variables, although it has some assumptions, such as linearity, independence of errors, homoscedasticity, and normality of errors, that need to be checked for the validity of the results.

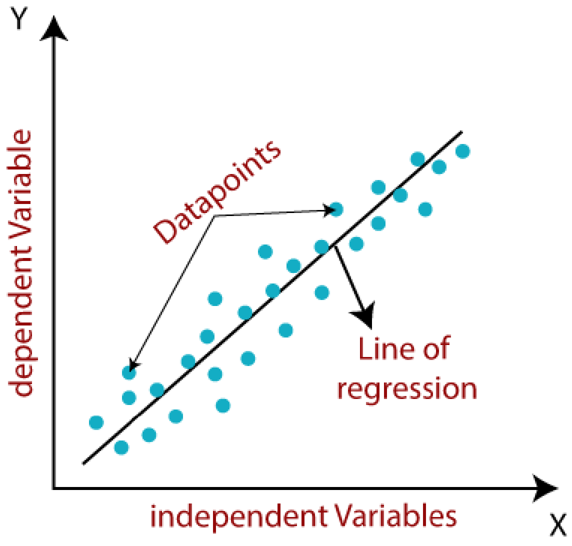


Figure 3.8: Simple Linear Regression

3.4.2 Random Forest Regression:

Random forest regression is a statistical method used for regression tasks, which is an extension of the basic concept of decision tree-based regression. It is a supervised machine learning algorithm that combines the predictions of multiple decision trees to make more accurate and robust predictions.

In random forest regression, a collection of decision trees, known as an ensemble, is created. Each decision tree is trained on a random subset of the training data, with replacement (also known as bootstrapping), and a random subset of the features (predictor variables) at each split. This introduces randomness into the model and helps to reduce overfitting, which is a common issue in decision tree-based models.

The main steps in building a random forest regression model are as follows:

Data preparation: Preprocess the data, including handling missing values, encoding categorical variables, and splitting the data into training and testing sets.

Ensemble creation: Create a collection of decision trees, typically by setting the number of trees (n_estimators) hyperparameter. Each tree is trained on a random subset of the training data.

Random feature selection: At each split of the decision tree, only a random subset of the features is considered, typically by setting the max_features hyperparameter. This helps to reduce the potential bias introduced by using all features in every tree.

Tree training: Train each decision tree using a criterion, such as mean squared error (MSE) or mean absolute error (MAE), to determine the optimal splits for the predictor variables.

Prediction: Once the random forest model is trained, it can be used to make predictions on the testing data by averaging the predictions of all the trees in the ensemble.

Random forest regression has several advantages, including its ability to handle non-linearity, high-dimensional data, and interactions between features. It is also less prone to overfitting compared to single decision tree models. Random forest regression is widely used in various fields, including finance, healthcare, marketing, and other areas where accurate predictions of numerical values are required. However, like any other machine learning algorithm, it also has some limitations, such as the potential for model interpretability, computational complexity, and hyperparameter tuning. Therefore, it is important to carefully evaluate and validate the performance of the random forest regression model on the specific data and problem at hand.

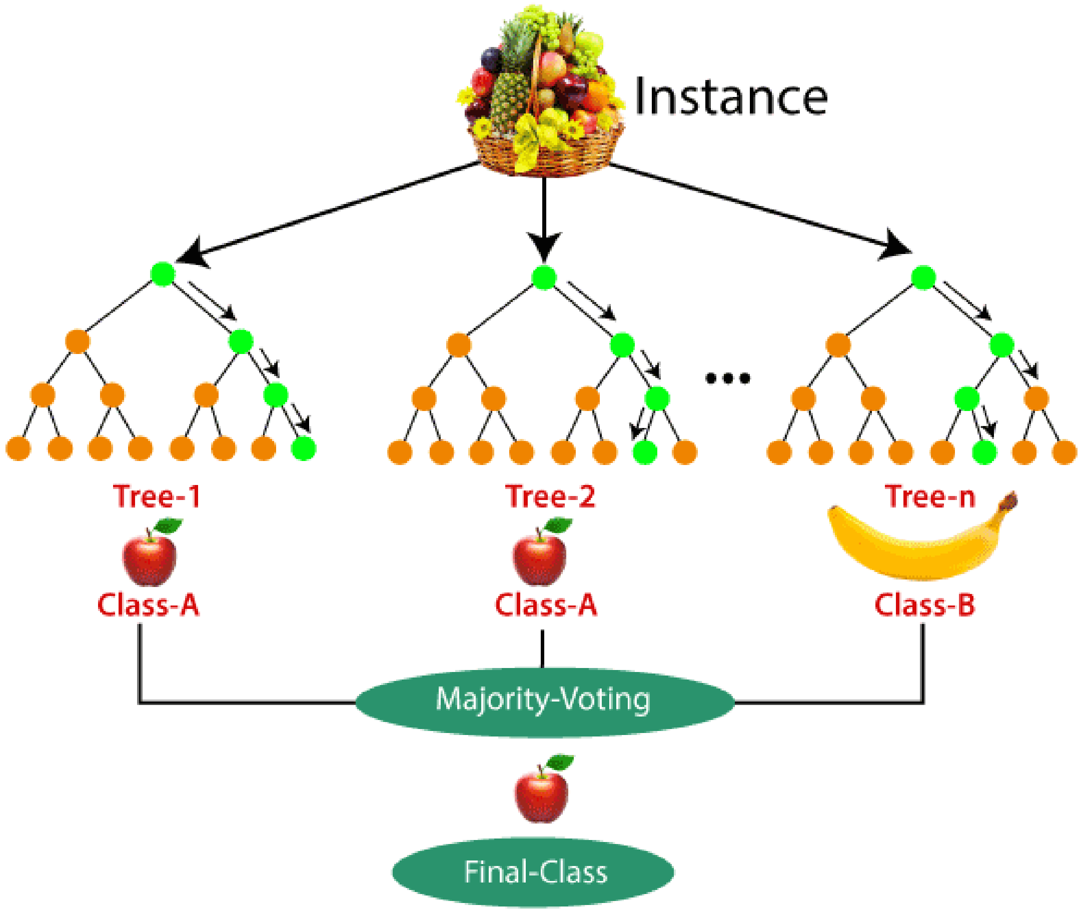


Figure 3.9: Random Forest Classifier

3.4.3 XGBoost Regression:

XGBoost (eXtreme Gradient Boosting) is a powerful and widely used gradient boosting framework for machine learning that can be used for regression tasks. It is an ensemble learning method that builds a predictive model by combining the predictions of multiple weak models in a weighted manner to make accurate predictions.

In XGBoost regression, an ensemble of decision trees is sequentially trained to correct the errors of the previous trees, with an emphasis on optimizing the gradient of the loss function. The main steps in building an XGBoost regression model are as follows:

Data preparation: Preprocess the data, including handling missing values, encoding categorical variables, and splitting the data into training and testing sets.

Model initialization: Initialize the XGBoost model with hyperparameters such as the learning rate (also known as the step size), the number of trees (n_estimators), and the maximum depth of the trees (max_depth).

Tree training: Train each decision tree in the ensemble using the gradient of the loss function to determine the optimal splits for the predictor variables. The model learns the optimal weights for the training instances, assigning higher weights to instances that were poorly predicted by the previous trees.

Regularization: Apply regularization techniques such as L1 (Lasso) or L2 (Ridge) regularization to control the complexity of the model and prevent overfitting. This is achieved through hyperparameters such as alpha and lambda.

Prediction: Once the XGBoost model is trained, it can be used to make predictions on the testing data by averaging the predictions of all the trees in the ensemble.

XGBoost regression has several advantages, including its ability to handle complex interactions between features, handle missing values, and robustness to outliers. It is also known for its speed and scalability, making it suitable for large datasets.

Additionally, XGBoost provides tools for feature selection and model interpretability, which can aid in understanding the importance of different features in the prediction.

However, like any other machine learning algorithm, XGBoost regression also has some limitations, such as the need for careful hyperparameter tuning, potential for overfitting, and potential lack of interpretability in very complex models. Therefore, it is important to thoroughly evaluate and validate the performance of the XGBoost regression model on the specific data and problem at hand.

3.5 HYPER PARAMETER TUNING

Hyperparameter tuning is an important step in machine learning model development that involves finding the optimal values for hyperparameters, which are parameters that are set prior to training a model and control the behavior of the learning algorithm. Hyperparameters can significantly impact the performance of a machine learning model, and tuning them properly can help improve the model's accuracy, generalization, and robustness.

Here are some common hyperparameters that may need tuning in machine learning models:

Learning rate: The learning rate controls the step size during model training. A higher learning rate can lead to faster convergence but may result in overshooting the optimal solution, while a lower learning rate can result in slower convergence. Tuning the learning rate can help find the right balance between convergence speed and accuracy.

Number of iterations or epochs: The number of iterations or epochs determines how many times the model goes through the training data during training. Too few iterations may result in underfitting, while too many iterations may result in overfitting. Tuning the number of iterations or epochs can help achieve the optimal trade-off between underfitting and overfitting.

Batch size: The batch size determines the number of training examples processed in one iteration during training. A smaller batch size can result in more frequent updates to the model weights, but can also increase the training time. A larger batch size can speed up training but may result in less frequent updates. Tuning the batch size can help find the right balance between computational efficiency and model performance.
Regularization strength: Regularization techniques such as L1 or L2 regularization are used to prevent overfitting by adding a penalty term to the model's loss function. The strength of regularization, controlled by hyperparameters such as regularization coefficient or alpha, needs to be tuned to find the optimal balance between model complexity and generalization.

Model architecture: Hyperparameters related to the model architecture, such as the number of layers, the number of neurons per layer, activation functions, etc., can also be tuned to find the best configuration for a specific problem. Different architectures may perform differently on different datasets, and tuning these hyperparameters can help optimize the model's performance.

There are several techniques that can be used for hyperparameter tuning, such as grid search, random search, Bayesian optimization, and more advanced techniques like genetic algorithms or neural architecture search. The choice of hyperparameter tuning technique may depend on the size of the dataset, the complexity of the model, and the available computational resources.

It's important to note that hyperparameter tuning is an iterative process and requires experimentation and evaluation of different hyperparameter values to find the best combination for a given machine learning problem. Proper hyperparameter tuning can significantly improve the performance of a machine learning model and make it more effective for real-world applications.

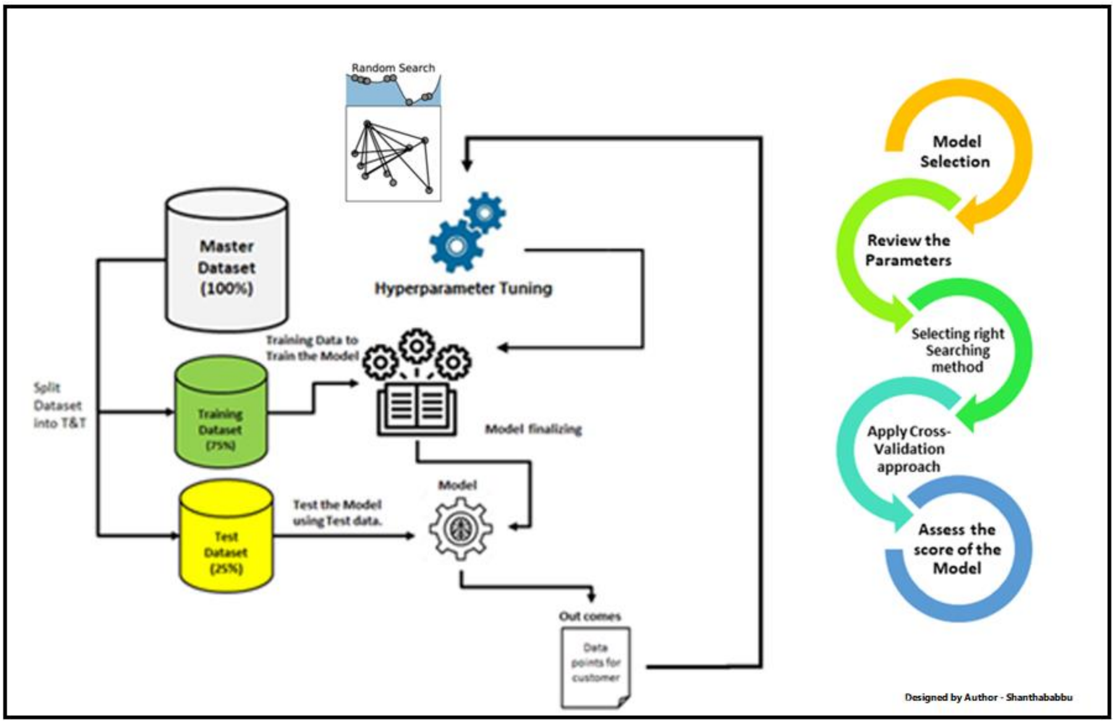


Figure 3.10: Hyper Parameter Tuning

3.6 PROPOSED SYSTEM

In proposed system we have assumed 3 branches of a company and manager of that respectedbranch can have access to its database of that branch to find hidden pattern in database. Using machine learning models, the proposed approach is used to anticipate sales.

The system architecture is depicted in Figure 1, when the model is used to train historical data from a shoppe provided in form of a CSV document & produces a prediction model using several ML algorithms such as linear regression, random forest, & XG Boost Regressor, before saving the model. The current data is tested with a csv file after the forecasting prototype is created. The forecast generates the projected value (RMSE, Mean, Standard, Minimum, Maximum).

Having the sales data of the retail store, the proposed work suggests the following various steps for predicting the sales of different categories available. The architectural diagram for the proposed algorithm is shown in Figure 4.9. The various steps involved are explained here under.

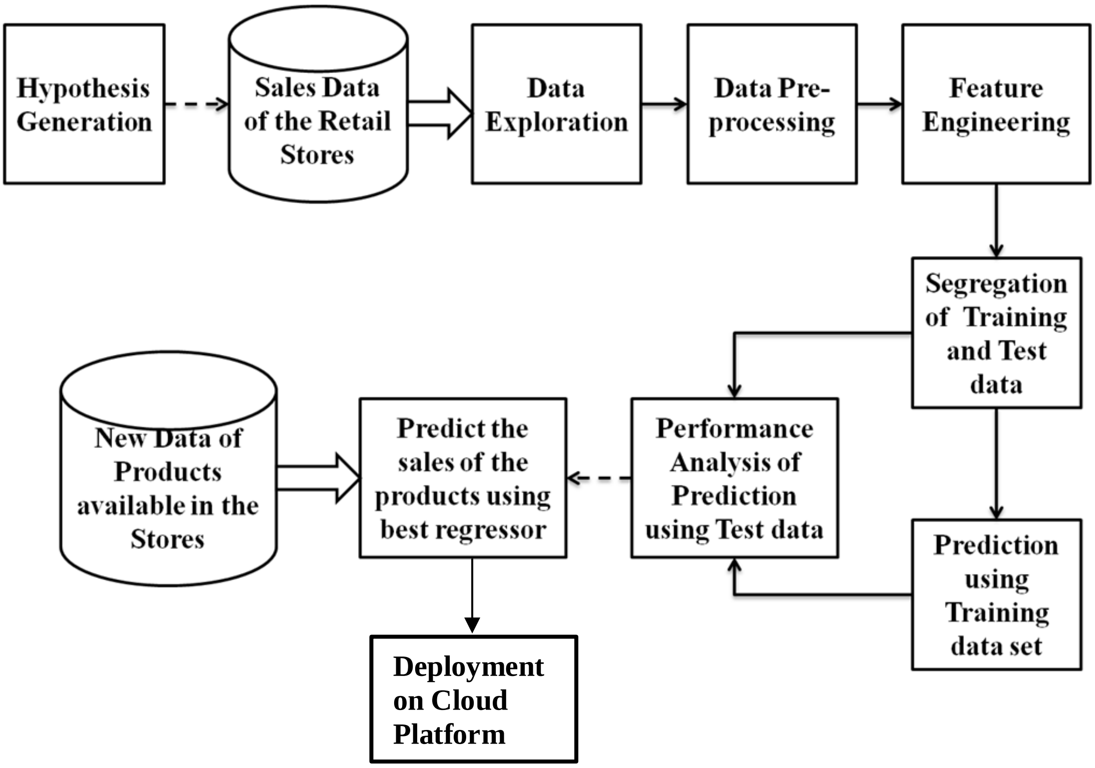

Hypothesis Generation
Sales Data of the Retail Stores
Data Exploration
Data Pre-processing
Feature Engineering
Segregation of Training and Test data
Prediction using Training data set
Performance Analysis of Prediction using Test data
Predict the sales of the products using best regressor
New Data of Products available in the Stores
Deployment on Cloud Platform

Figure 3.11: Proposed System

CHAPTER 4
IMPLEMENTATION

4.1 IMPLEMENTATION

We have gone through various stages of implementation in our project. The project can be broken in many parts and each part have an equal importance. When we thought of the project “Deploying Machine Learning Model on Cloud for Supermarket Sales Data Analytics and Forecasting” we read many research paper (Reference to it can be found in Literature Survey chapter) and wefound that the work done on this topic is very less and also no exact algorithm is defined to improve accuracy.

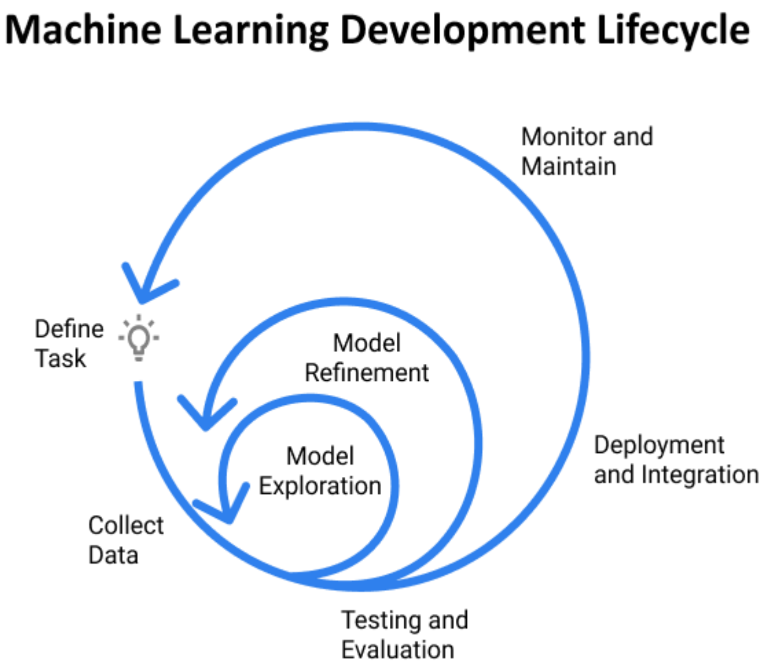


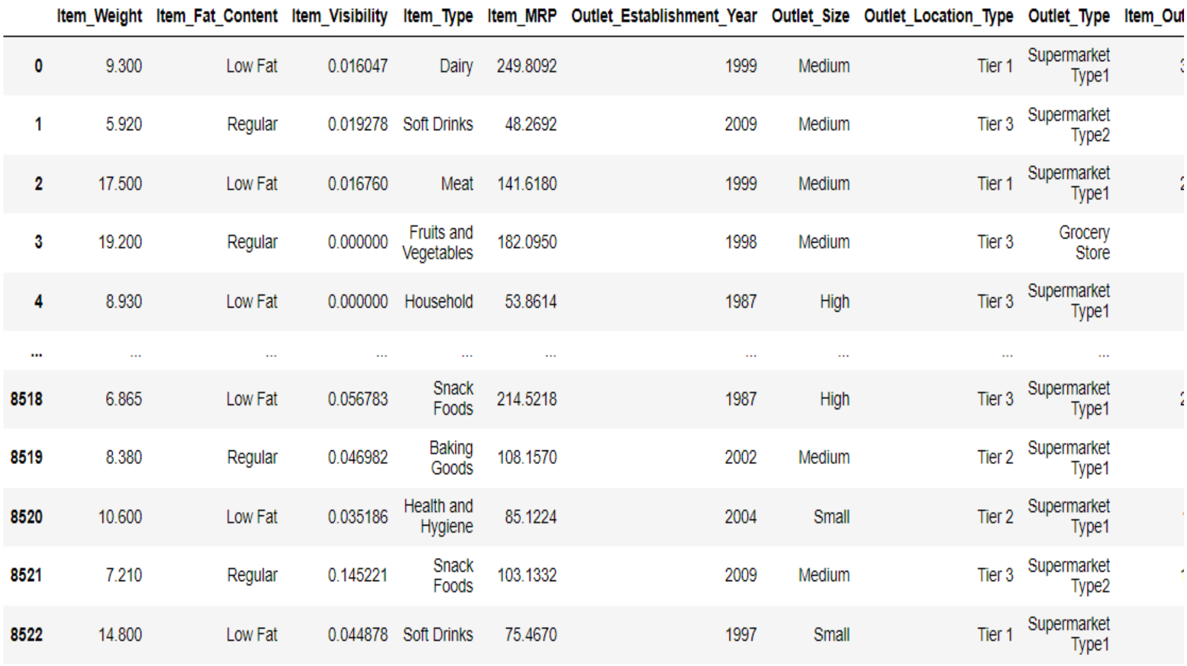
Figure 4.1: Machine Learning Life Cycle

The various stages our project has gone through are:

- Data Collection
- Data Exploration
- Data Pre-processing
- Data Visualization

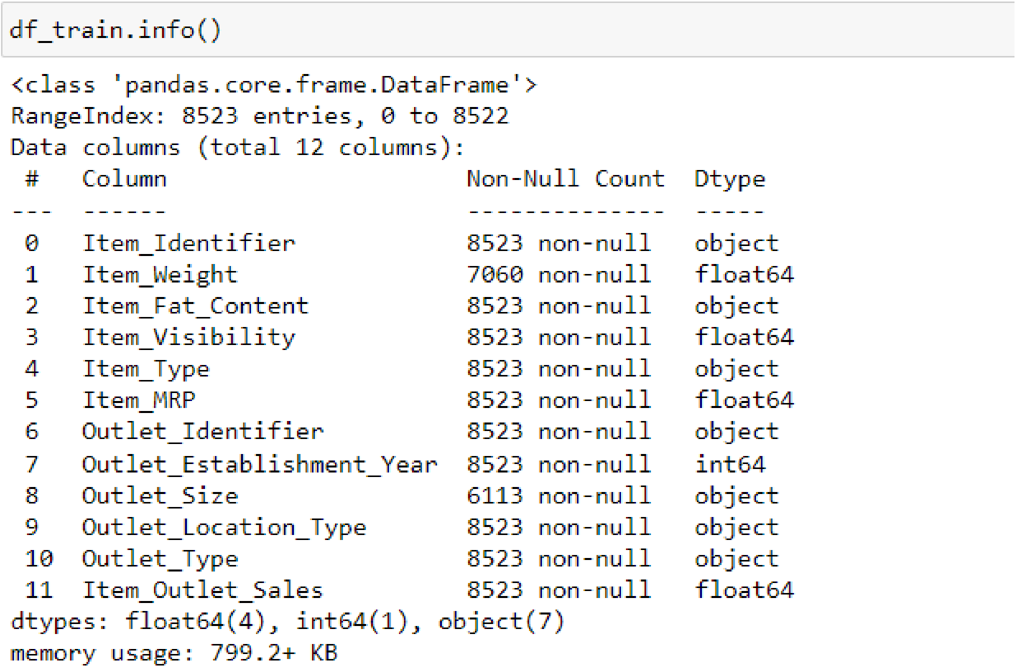
4.1.1 Data Collection

The initial step was Data is gathered in the form of datasets from the company's database or from several data warehousing websites like Kaggle, UCI, and Analytics Vidhya. We gathered data about Big Mart's transactions from its 10 locations, each of which has a stock of 1559 different items. By combining all of these facts, it is feasible to pinpoint the roles that various aspects of a commodity play and how those aspects affect its transactions. Machine learning is impossible without data, which is why data is so important. It requires data in one form or the other. Just like we humans need food for our developmentof mind and then when we get another type of data by visualizing, hearing, etc., and get experience from such data. That data plays a vital role in the typeof human we will be in the future. Utilizing whole these findings, it is possible to identify what part-specific characteristics of a commodity play and by virtue of what they influence its transactions.



	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Type	Outlet_Type	Item_Out
0	9.300	Low Fat	0.016047	Dairy	249.8092	1999	Medium	Tier 1	Supermarket Type1	
1	5.920	Regular	0.019278	Soft Drinks	48.2692	2009	Medium	Tier 3	Supermarket Type2	
2	17.500	Low Fat	0.016760	Meat	141.6180	1999	Medium	Tier 1	Supermarket Type1	
3	19.200	Regular	0.000000	Fruits and Vegetables	182.0950	1998	Medium	Tier 3	Grocery Store	
4	8.930	Low Fat	0.000000	Household	53.8614	1987	High	Tier 3	Supermarket Type1	
...	...	...	...	...	...	...	...	...	...	
8518	6.865	Low Fat	0.056783	Snack Foods	214.5218	1987	High	Tier 3	Supermarket Type1	
8519	8.380	Regular	0.046982	Baking Goods	108.1570	2002	Medium	Tier 2	Supermarket Type1	
8520	10.600	Low Fat	0.035186	Health and Hygiene	85.1224	2004	Small	Tier 2	Supermarket Type1	
8521	7.210	Regular	0.145221	Snack Foods	103.1332	2009	Medium	Tier 3	Supermarket Type2	
8522	14.800	Low Fat	0.044878	Soft Drinks	75.4670	1997	Small	Tier 1	Supermarket Type1	

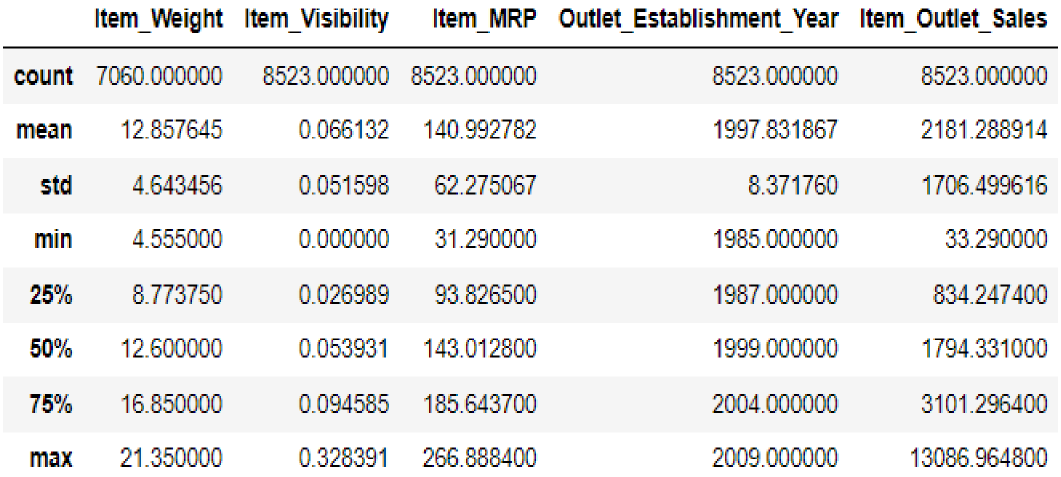
Figure 4.2: Screenshot of Dataset



```
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 12 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Item_Identifier            8523 non-null   object 
 1   Item_Weight                7060 non-null   float64
 2   Item_Fat_Content           8523 non-null   object 
 3   Item_Visibility            8523 non-null   float64
 4   Item_Type                  8523 non-null   object 
 5   Item_MRP                   8523 non-null   float64
 6   Outlet_Identifier          8523 non-null   object 
 7   Outlet_Establishment_Year  8523 non-null   int64  
 8   Outlet_Size                6113 non-null   object 
 9   Outlet_Location_Type       8523 non-null   object 
 10  Outlet_Type                8523 non-null   object 
 11  Item_Outlet_Sales          8523 non-null   float64
dtypes: float64(4), int64(1), object(7)
memory usage: 799.2+ KB
```

Figure 4.3: Datatypes used within Dataset



	Item_Weight	Item_Visibility	Item_MRP	Outlet_Establishment_Year	Item_Outlet_Sales
count	7060.000000	8523.000000	8523.000000	8523.000000	8523.000000
mean	12.857645	0.066132	140.992782	1997.831867	2181.288914
std	4.643456	0.051598	62.275067	8.371760	1706.499616
min	4.555000	0.000000	31.290000	1985.000000	33.290000
25%	8.773750	0.026989	93.826500	1987.000000	834.247400
50%	12.600000	0.053931	143.012800	1999.000000	1794.331000
75%	16.850000	0.094585	185.643700	2004.000000	3101.296400
max	21.350000	0.328391	266.888400	2009.000000	13086.964800

Figure 4.4: Dataset description using describe() function.

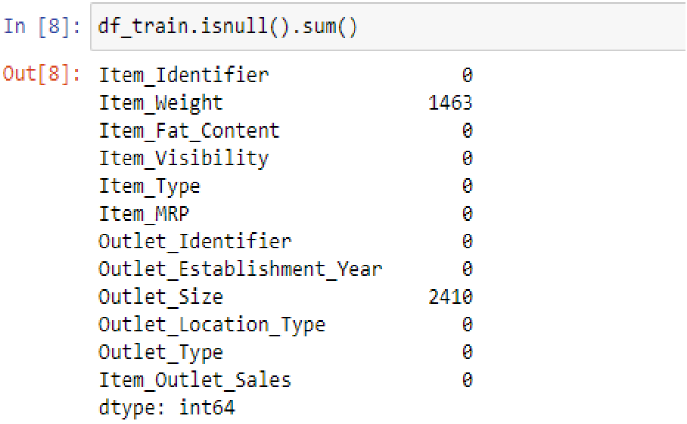
4.1.2 Data Exploration

Data exploration is the process of analyzing and understanding a dataset to extract useful information and insights. It involves summarizing the main features of the dataset, identifying patterns and trends, and detecting outliers or anomalies. Data exploration is an important step in the data analysis process, as it helps to guide the selection of appropriate statistical methods and models, and to identify potential problems or limitations in the data. Some common techniques used in data exploration include data visualization, summary statistics, and exploratory data analysis (EDA). Data visualization can help to reveal patterns and relationships in the data, such as scatter plots, histograms, and box plots. Summary statistics, such as mean, median, and standard deviation, can provide a quick overview of the distribution of the data. EDA involves the use of statistical techniques, such as regression analysis and hypothesis testing, to explore relationships between variables and to identify important predictors of the outcome variable.

4.1.3 Data Pre-Processing

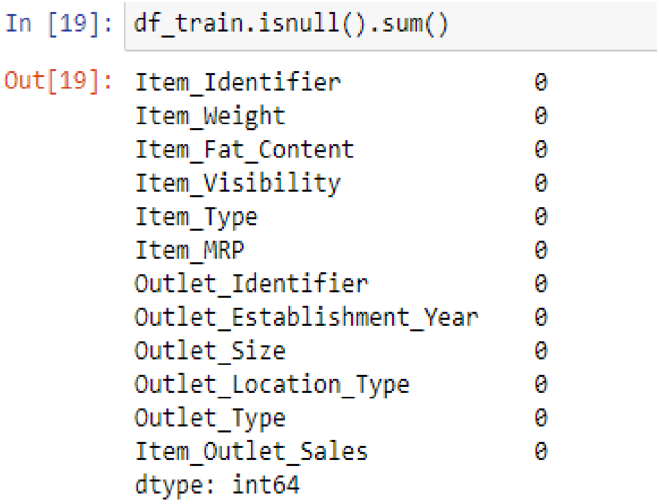
Big Mart’s data scientists collected sales data of their 10 stores situated at different locations with each store having 1559 different products as per 2013 data collection. Using all the observations it is inferred what role certain properties of an item play and how they affect their sales.

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 item fat 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.



```
In [8]:  df_train.isnull().sum()

Out[8]:  Item_Identifier                 0
         Item_Weight                  1463
         Item_Fat_Content                0
         Item_Visibility                 0
         Item_Type                       0
         Item_MRP                        0
         Outlet_Identifier               0
         Outlet_Establishment_Year       0
         Outlet_Size                  2410
         Outlet_Location_Type            0
         Outlet_Type                     0
         Item_Outlet_Sales               0
         dtype: int64
```

Figure 4.5: Before Data Pre-processing



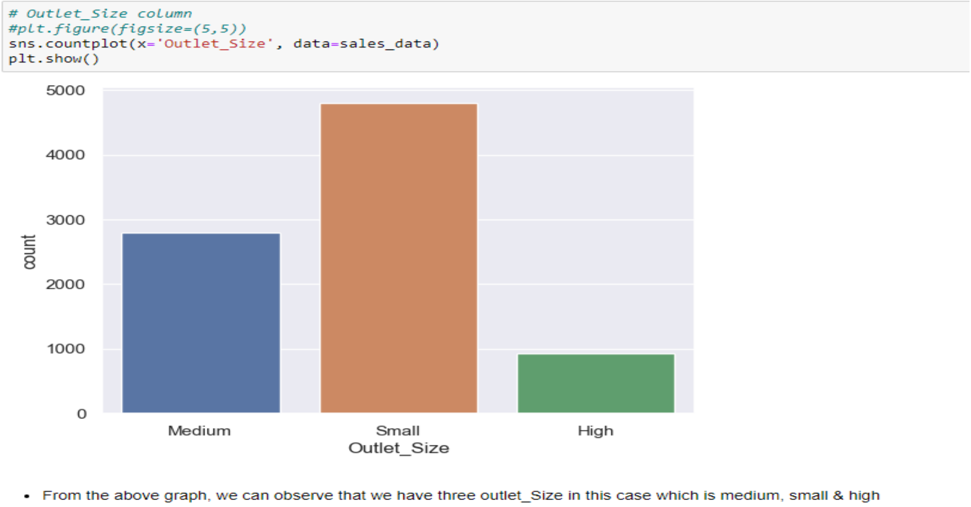
```
In [19]:  df_train.isnull().sum()

Out[19]:  Item_Identifier              0
          Item_Weight                  0
          Item_Fat_Content             0
          Item_Visibility              0
          Item_Type                    0
          Item_MRP                     0
          Outlet_Identifier            0
          Outlet_Establishment_Year    0
          Outlet_Size                  0
          Outlet_Location_Type         0
          Outlet_Type                  0
          Item_Outlet_Sales            0
          dtype: int64
```

Figure 4.6: After Data Pre-Processing

The data is then randomized, which eliminates the influence of the sequence in which we acquired and/or changed collected data, as seen in Figure 4.6.

4.1.2 Data Visualization

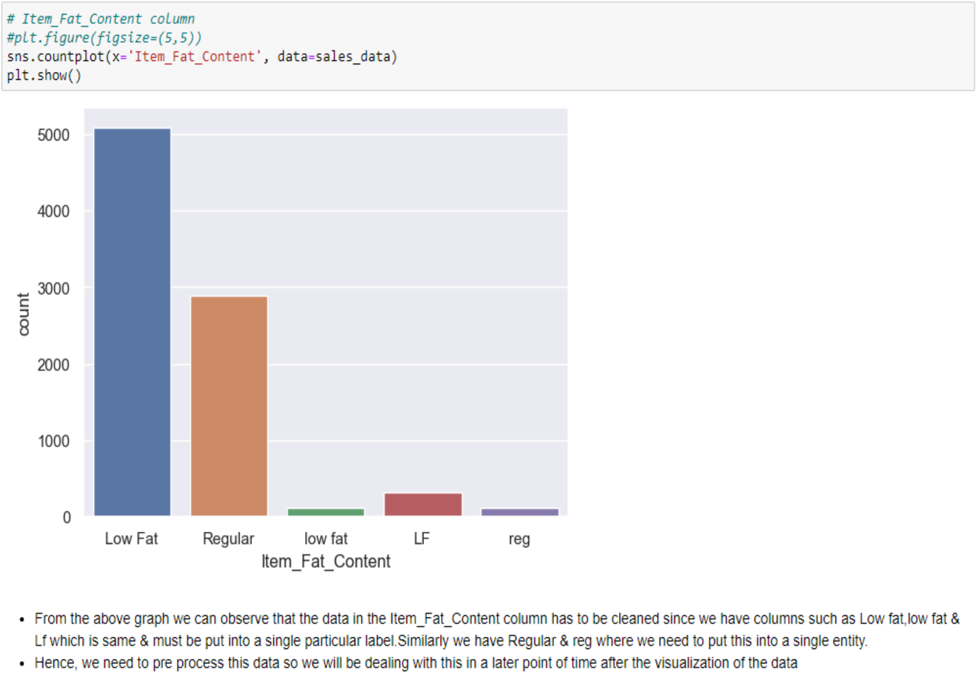
Next step is visualizing the info help to discover contextual connections between variables or class imbalances (bias warnings!) or execute additional experimental findings. After that, the data is therefore divided into preparation and experiment judgment sets as proved.



```
# Outlet_Size column
#plt.figure(figsize=(5,5))
sns.countplot(x='Outlet_Size', data=sales_data)
plt.show()
```

- From the above graph, we can observe that we have three outlet_Size in this case which is medium, small & high

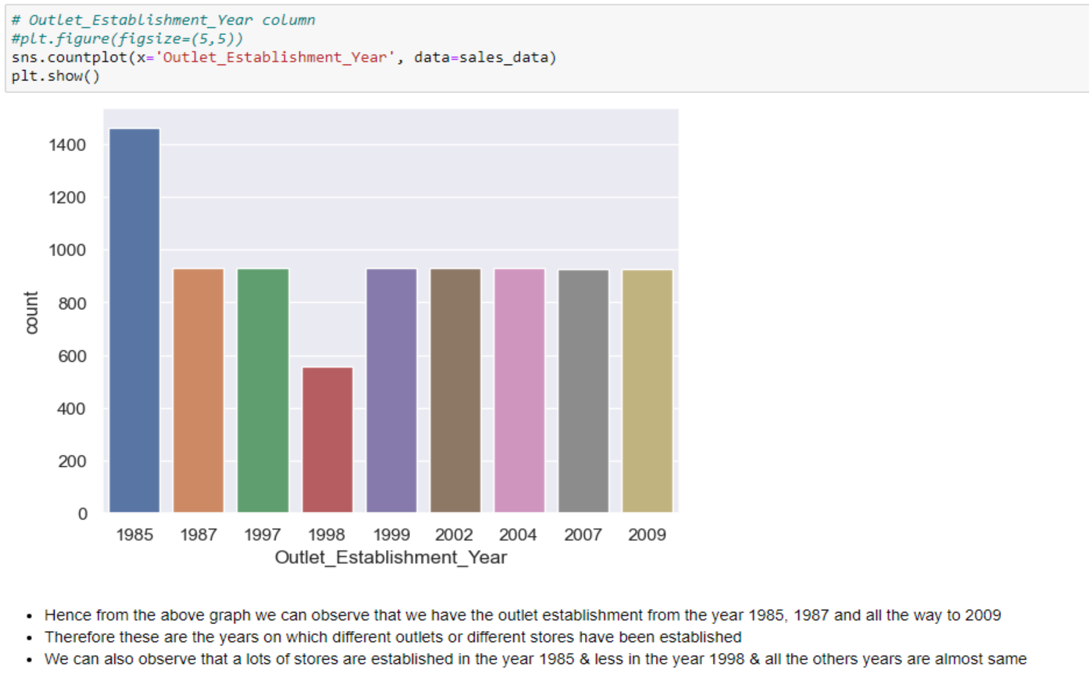
Figure 4.7: Visualization of outlet_Size using countplot



```
# Item_Fat_Content column
#plt.figure(figsize=(5,5))
sns.countplot(x='Item_Fat_Content', data=sales_data)
plt.show()
```

- From the above graph we can observe that the data in the Item_Fat_Content column has to be cleaned since we have columns such as Low fat,low fat & Lf which is same & must be put into a single particular label.Similarly we have Regular & reg where we need to put this into a single entity.
- Hence, we need to pre process this data so we will be dealing with this in a later point of time after the visualization of the data

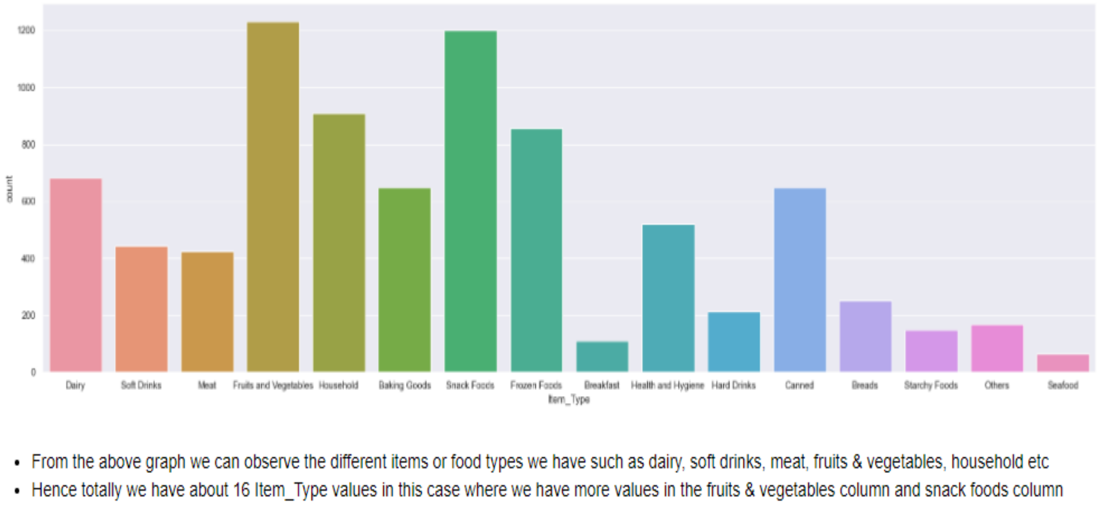
Figure 4.8: Visualization of Item_Fat_Content using countplot



```
# Outlet_Establishment_Year column
#plt.figure(figsize=(5,5))
sns.countplot(x='Outlet_Establishment_Year', data=sales_data)
plt.show()
```

- Hence from the above graph we can observe that we have the outlet establishment from the year 1985, 1987 and all the way to 2009
- Therefore these are the years on which different outlets or different stores have been established
- We can also observe that a lots of stores are established in the year 1985 & less in the year 1998 & all the others years are almost same

Figure 4.9: Visualization of Outlet_Establishment_Year using countplot



- From the above graph we can observe the different items or food types we have such as dairy, soft drinks, meat, fruits & vegetables, household etc
- Hence totally we have about 16 Item_Type values in this case where we have more values in the fruits & vegetables column and snack foods column

Figure 4.10: Visualized data of available items in dataset

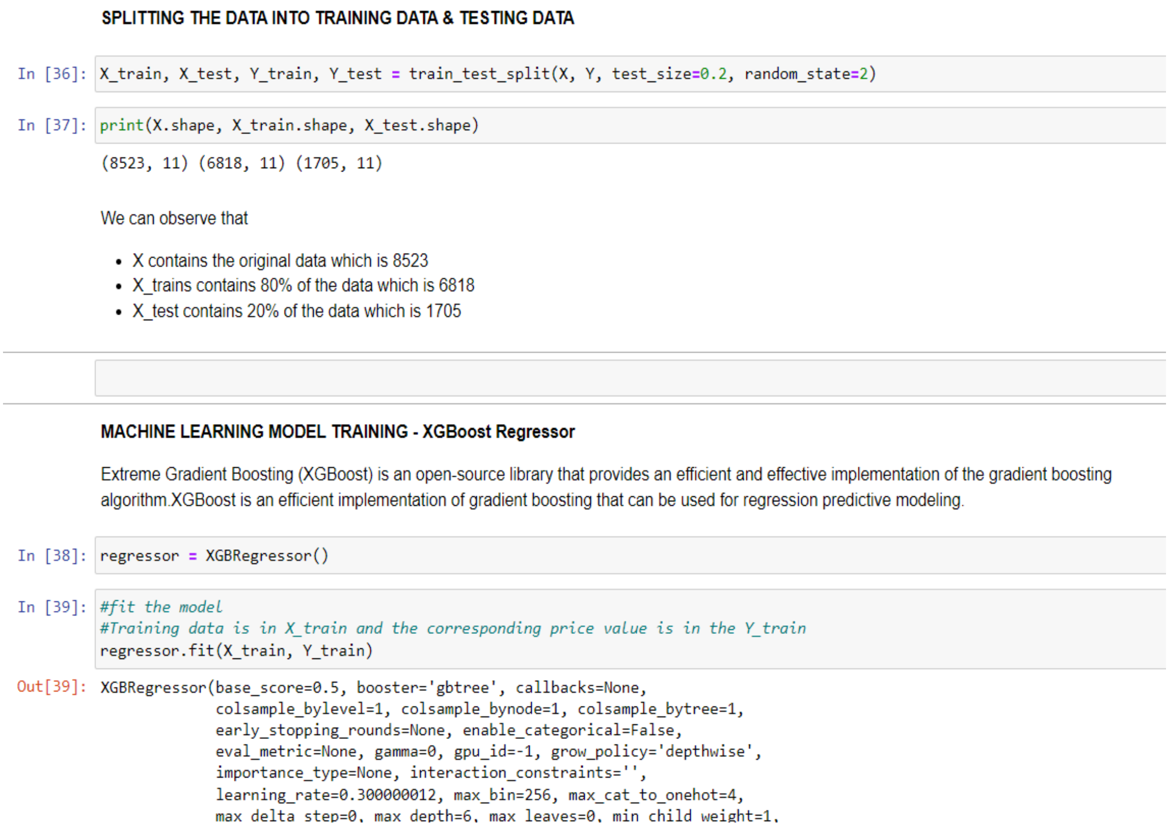
CHAPTER 5
DEPLOYMENT

5.1 Training the Model

A training model is a dataset that is used to train an ML algorithm. It consists of the sample output data and the corresponding sets of input data that have an influence on the output. The training model is used to run the input data through the algorithm to correlate the processed output against the sample output. The result from this correlation is used to modify the model.

Model training in machine language is the process of feeding an ML algorithm with data to help identify and learn good values for all attributes involved.

The info is separated into two categories: training and testing after using suitable algorithmsfor model building the data is then trained and tested for getting the desired outcome shown in Figure 5.1.



SPLITTING THE DATA INTO TRAINING DATA & TESTING DATA

```
In [36]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)

In [37]: print(X.shape, X_train.shape, X_test.shape)

         (8523, 11) (6818, 11) (1705, 11)
```

We can observe that

- X contains the original data which is 8523
- X_trains contains 80% of the data which is 6818
- X_test contains 20% of the data which is 1705

MACHINE LEARNING MODEL TRAINING - XGBoost Regressor

Extreme Gradient Boosting (XGBoost) is an open-source library that provides an efficient and effective implementation of the gradient boosting algorithm.XGBoost is an efficient implementation of gradient boosting that can be used for regression predictive modeling.

```
In [38]: regressor = XGBRegressor()

In [39]: #fit the model
         #Training data is in X_train and the corresponding price value is in the Y_train
         regressor.fit(X_train, Y_train)

Out[39]: XGBRegressor(base_score=0.5, booster='gbtree', callbacks=None,
                      colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
                      early_stopping_rounds=None, enable_categorical=False,
                      eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
                      importance_type=None, interaction_constraints='',
                      learning_rate=0.300000012, max_bin=256, max_cat_to_onehot=4,
                      max_delta_step=0, max_depth=6, max_leaves=0, min_child_weight=1,
```

Figure 5.1: Model splitting & Training

5.1.1 Evaluating the Model

- To evaluate a machine learning model, there are several metrics and techniques that can be used depending on the type of problem being solved and the nature of the data. Here are some of the commonly used evaluation metrics:

- **Accuracy:** It is a basic metric that measures the proportion of correct predictions made by the model. However, it can be misleading in cases where the dataset is imbalanced.

- **Precision and Recall:** Precision measures how many of the predicted positive examples are actually positive. Recall measures how many of the actual positive examples were predicted correctly. These metrics are useful when the dataset is imbalanced.

- **F1 Score:** It is the harmonic mean of precision and recall. It is a useful metric when you want to balance between precision and recall.

- **ROC Curve:** ROC (Receiver Operating Characteristic) curve plots the true positive rate against the false positive rate for different threshold values. It helps to understand the performance of the model at different threshold values.

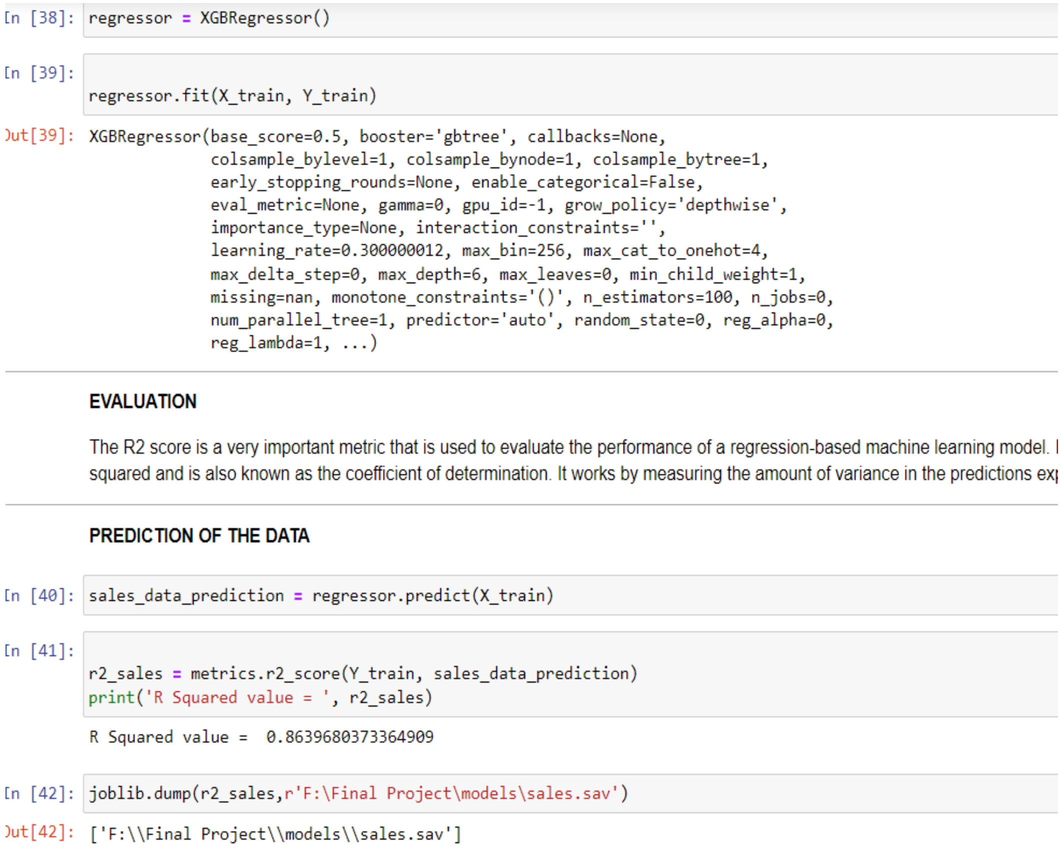
- **Confusion Matrix:** A confusion matrix is a table that shows the predicted classes versus the actual classes. It is useful to understand the number of correct and incorrect predictions made by the model.

- **Cross-validation:** Cross-validation is a technique used to evaluate the performance of the model by training and testing on different subsets of the data. It helps to prevent overfitting and gives a more accurate estimate of the model's performance.

- It is important to choose the right metric(s) based on the problem at hand and the characteristics of the data. Additionally, it is always a good practice to compare the performance of different models using the same evaluation metrics to choose the best one.

- This hidden information is intended to be indicative of predictive accuracy in the actual world, but it may still be used to fine-tune models (as opposed to test data, which does not).

- Relevant to the topic, access to data, dataset details, and other factors, a suitable train or eval split of 80 or 20, 70 or 30, or something similar is recommended.



```
[n [38]: regressor = XGBRegressor()

[n [39]:
         regressor.fit(X_train, Y_train)

)ut[39]: XGBRegressor(base_score=0.5, booster='gbtree', callbacks=None,
                      colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
                      early_stopping_rounds=None, enable_categorical=False,
                      eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
                      importance_type=None, interaction_constraints='',
                      learning_rate=0.300000012, max_bin=256, max_cat_to_onehot=4,
                      max_delta_step=0, max_depth=6, max_leaves=0, min_child_weight=1,
                      missing=nan, monotone_constraints='()', n_estimators=100, n_jobs=0,
                      num_parallel_tree=1, predictor='auto', random_state=0, reg_alpha=0,
                      reg_lambda=1, ...)
```

EVALUATION

The R2 score is a very important metric that is used to evaluate the performance of a regression-based machine learning model. I squared and is also known as the coefficient of determination. It works by measuring the amount of variance in the predictions ex

PREDICTION OF THE DATA

```
[n [40]: sales_data_prediction = regressor.predict(X_train)

[n [41]:
         r2_sales = metrics.r2_score(Y_train, sales_data_prediction)
         print('R Squared value = ', r2_sales)

         R Squared value =  0.8639680373364909

[n [42]: joblib.dump(r2_sales,r'F:\Final Project\models\sales.sav')

)ut[42]: ['F:\\Final Project\\models\\sales.sav']
```

Figure 5.2: Evaluation of the model

5.2 Deployment

- In this project we have built the model which is transformed into websites or web applications or in any other form according to client's requirements.

- Here we are going to make a web application on the dataset given to us after buildingthe ML model. The deployment is under the assumption that the person has a fair knowledge of running python code and is familiar with simple ML libraries like Sci-kit Learn, Pandas, NumPy.

- We have used **Flask** which is a Python based microframework used for developing small scale websites. Flask is very easy to make Restful API's using python. As of now, we have developed a model i.e **model.sav** which can predict a class of the data based ona various attribute of the data. We have designed a web application where the user will input all the attribute values and the data will be given the model, based on the training given to the model, the model will predict, what are the sales of the item or category of items whose details has been fed. Then use random forest model and linear regression topredict the sales.

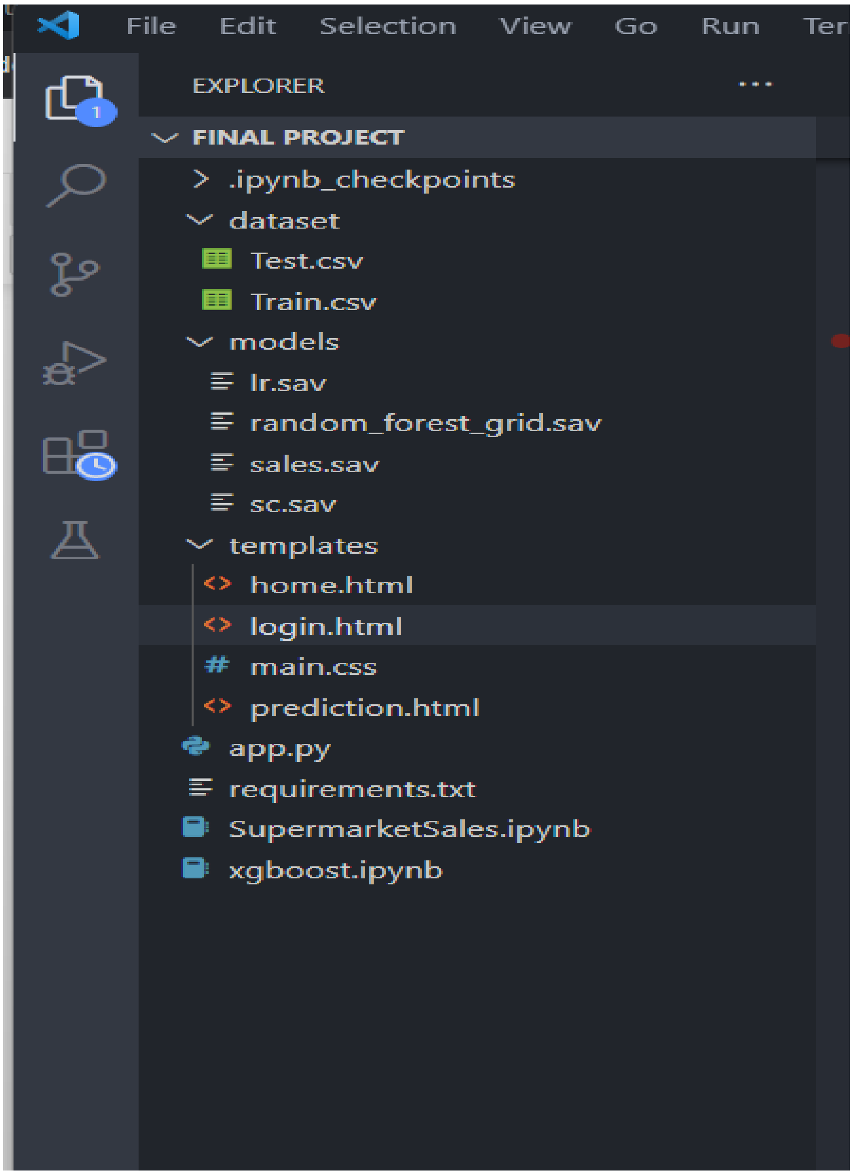
- **Flask script** – Before starting with the coding part, we need to download flask and some other libraries. Here, we make use of virtual environment, where all the libraries are managed and makes the development job easier.

- Here we import the libraries, then using **app=Flask(_name__)** we create an instance of **flask. @app.route('/')** is used to tell flask what **URL** should trigger the **function index()** and in the function index we use **render_template('index.html')** to display the script **index.html** in the browser.

- This should run the application and launch a simple server. **Open http://127.0.0.1:9457/** to see the html form.

- **HTML Form –** In order to collect the data, we created html form which would contain all the different options to select from each attribute. Here, we have created a simple form using html.

When we click on the predict button in index.html, it predicts the salary for the values entered by the user (3 inputs), then passes on the variable `Y_pred `outputted from the model and sends it back to index.html template as “$ {{'%0.2f' format(prediction|float)}}”



```
FINAL PROJECT
  > .ipynb_checkpoints
  ∨ dataset
      Test.csv
      Train.csv
  ∨ models
      lr.sav
      random_forest_grid.sav
      sales.sav
      sc.sav
  ∨ templates
      home.html
      login.html
      main.css
      prediction.html
    app.py
    requirements.txt
    SupermarketSales.ipynb
    xgboost.ipynb
```

Figure 5.3: Directory Structure

5.2.1 Deployment on Amazon Web Services using EC2 instance

- Launch an EC2 instance: Create an EC2 instance with the desired configuration, such as the instance type, operating system, and storage. You can do this using the AWS Management Console, AWS CLI, or SDKs.

- Connect to the EC2 instance: Once the EC2 instance is launched, you can connect to it using SSH (Secure Shell) from your local machine or another remote machine. You'll need the key pair associated with the instance for authentication.

- Install the necessary dependencies: Install any dependencies required for your machine learning model, such as Python, scikit-learn, joblib, and any other libraries or packages.

- Upload your machine learning model: Transfer your serialized machine learning model (e.g., the joblib file) to the EC2 instance. You can use tools like SCP or SFTP to upload the file to the instance.

- Load and use the machine learning model: Load the serialized model into your Python code running on the EC2 instance using joblib or other appropriate libraries. You can then use the model to make predictions on new data.

- Expose the model via an API: If you want to expose your machine learning model as an API, you can use a web framework like Flask or Django to create a RESTful API that can receive HTTP requests and invoke your machine learning model to make predictions.

- Configure security: Ensure that appropriate security measures are in place, such as securing the EC2 instance with proper access controls, setting up SSL encryption for API requests, and implementing other security best practices.

5.3 Model Building for Prediction

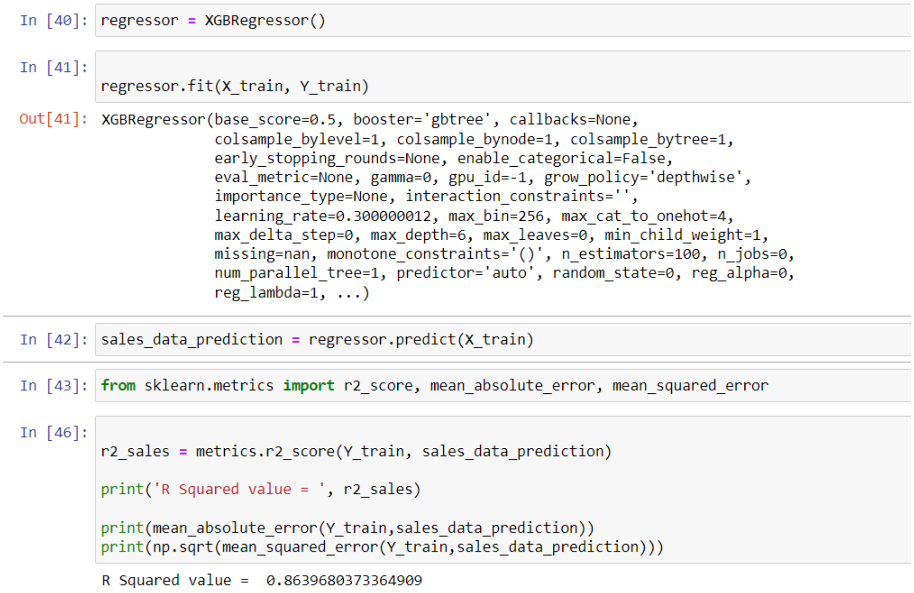
After the dataset is split into training and testing sets, the training set is fed into the algorithm so that it can learn how to predict the values. Various regression algorithms like Linear Regression, Random Forest Regression, XGBoost Regression, etc. have been applied. Along with these boosting algorithms like XGBoost has also been applied to the dataset for increasing the accuracy.

5.4 Prediction Results

- Item_MRP optimizes Maximum Outlet sales (positive correlation with the target).

- Linear Regression has lower accuracy than Random Forest Regressor algorithm. Linear regression accuracy came to be 50% and Random Forest accuracy came near 54%.

- Also, model accuracy and score of XGboost model can reach nearly 86% if built with more hypothesis consideration and analysis, as shown by code snippet in Figure 5.4.



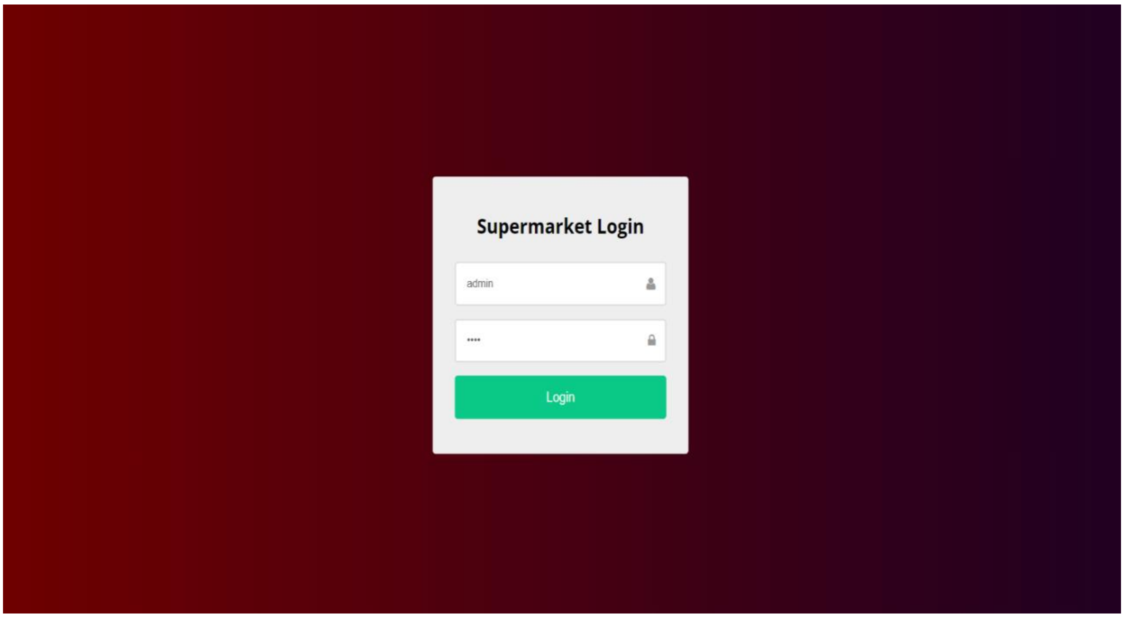
```
In [40]: regressor = XGBRegressor()

In [41]:
         regressor.fit(X_train, Y_train)

Out[41]: XGBRegressor(base_score=0.5, booster='gbtree', callbacks=None,
                      colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
                      early_stopping_rounds=None, enable_categorical=False,
                      eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
                      importance_type=None, interaction_constraints='',
                      learning_rate=0.300000012, max_bin=256, max_cat_to_onehot=4,
                      max_delta_step=0, max_depth=6, max_leaves=0, min_child_weight=1,
                      missing=nan, monotone_constraints='()', n_estimators=100, n_jobs=0,
                      num_parallel_tree=1, predictor='auto', random_state=0, reg_alpha=0,
                      reg_lambda=1, ...)

In [42]: sales_data_prediction = regressor.predict(X_train)

In [43]: from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error

In [46]:
         r2_sales = metrics.r2_score(Y_train, sales_data_prediction)

         print('R Squared value = ', r2_sales)

         print(mean_absolute_error(Y_train,sales_data_prediction))
         print(np.sqrt(mean_squared_error(Y_train,sales_data_prediction)))

         R Squared value =  0.8639680373364909
```

Figure 5.4: Code showing model score of XGBoost Model

Algorithms	Training to Testing Ratio					
	80:20		75:25		70:30	
	RMSE	R^2	RMSE	R^2	RMSE	R^2
Linear Regression	1162.44	0.50	1168.35	0.50	1179.45	0.50
Random Forest	1107.87	0.54	1123.43	0.54	1130.57	0.55
XGBoost	624.50	0.86	631.66	0.86	643.24	0.86

Table 2: Simulation Results of Various Algorithms

Result:

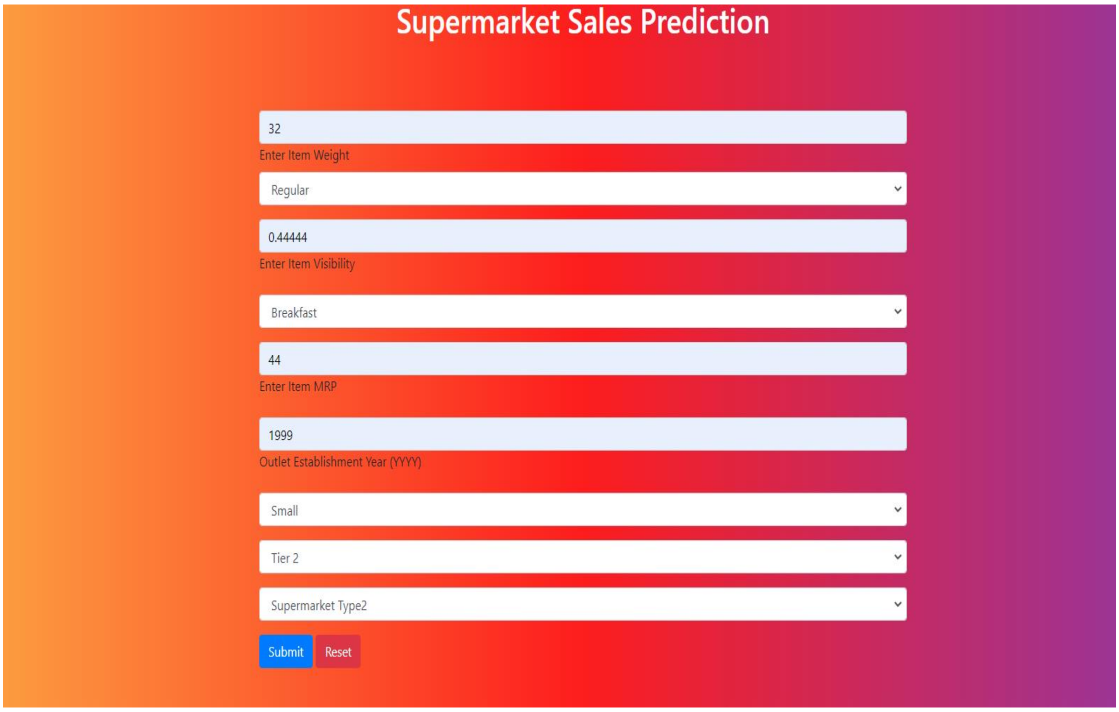


Supermarket Login

admin

••••

Snapshot 1: Basic Login Page for admin



Supermarket Sales Prediction

32

Enter Item Weight

Regular

0.44444

Enter Item Visibility

Breakfast

44

Enter Item MRP

1999

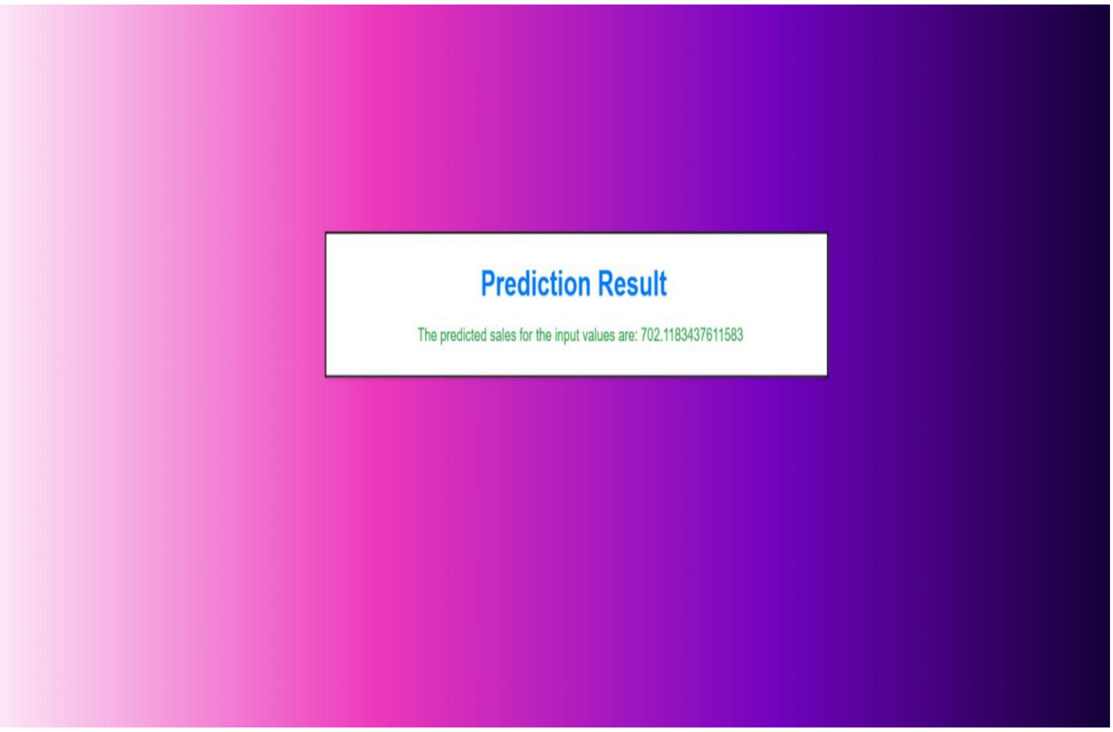
Outlet Establishment Year (YYYY)

Small

Tier 2

Supermarket Type2

Snapshot 2: Prediction System



Prediction Result

The predicted sales for the input values are: 702.1183437611583

Snapshot 3: Window Displaying Output

CHAPTER 6
CONCLUSION

CONCLUSION

Sales forecasting is essential in every business industry. Sales revenue analysis will assist in obtaining the facts required to anticipate both revenue and profits using sales forecasts. On supermarket sales data, various Machine Learning approaches such as Simple Linear Regression, Random Forest Regression, XGBoost Regression and Hyperparameter Tuning were assessed to identify the essential elements influencing sales and propose a solution for projecting sales. Following the application of measurements such as accuracy, mean absolute error, and maximum error, the Random Forest Regression is determined to be the optimal algorithm based on the obtained data, thereby accomplishing the goal of this thesis.

Because of these factors, it is critical to design productive models in order to deliver robust and accurate findings. Simultaneously, the fields and attributes used in this research were insufficient for future investigation. It was the most difficult challenge we encountered during the research. We have, however, thoroughly weighed our work by implementing effective ML algorithms for prediction and forecasting. The current research can be accelerated by employing Big Data as a predictive analytics tool in sales forecasting. Big data analysis and forecasting are regarded as critical disciplines in modern business.

REFERENCES

REFERENCES

[1] Bohdan M Pavlyshenko. Machine-learning models for sales time series forecast- ing. *Data,* 4(1):15, 2019.

[2] Akshay Krishna, Akhilesh V, Animikh Aich, Chetana Hegde, "Sales-forecasting of Retail Stores using Machine Learning Techniques", in 3rd IEEE International Conference on Computational Systems and Information Technology for Sustainable Solutions 2018.

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

[4] Krzysztof J Cios, Witold Pedrycz, Roman W Swiniarski, and Lukasz Andrzej Kurgan. Data mining: a knowledge discovery approach. Springer Science & Business Media, 2007. 32 References 33

[5] 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.

[6] Sigrist, F., & Hirnschall, C. (2018). Gradient Tree-Boosted Tobit Models for Default Prediction.

[7] Daelemans, W., Hoste, V., De Meulder, F., Naudts, B. et al., (Eds.). 2003. Combined optimization of feature selection and algorithm parameters in machine learning of language. European Conference on Machine Learning, ECML 2003. LNCS, pp. 84–95.

[8] Stearns, B., Rangel, F., Rangel, F., de Faria, F. F., Oliveira, J., & Ramos, A. A. D. S. (2017). Scholar Performance Prediction using Boosted Regression Trees Techniques. In European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN). Citeseer.

[9] Gangadhar Shobha and Shanta Rangaswamy, "Machine Learning", R. V. College of Engineering, Bengaluru, India.

[10] Sunitha Cheriyan, Shaniba Ibrahim, Saju Mohanan, Susan Treesa. "Intelligent Sales Prediction Using Machine Learning Techniques" in proceedings of **2018 iCCECE Conference.**

[11] Nikita Malik, Karan Singh "SALES PREDICTION MODEL FOR BIG MART" article published in ResearchGate in proceedings of July 2020.

[16] Wangchao Lou, Xiaoqing Wang, Fan Chen, Yixiao Chen, Bo Jiang, and Hua Zhang. Sequence based prediction of dna-binding proteins based on hybrid feature selection using random forest and gaussian naive bayes. *PloS one*, 9(1), 2014.

[17] Taiwo Oladipupo Ayodele. Types of machine learning algorithms. *New advances in machine learning*, pages 19–48, 2010.

[18] Sanford Weisberg. *Applied linear regression*, volume 528. John Wiley & Sons, 2005.

[19] Gradient Boosting documentation. https://turi.com/learn/userguide/ supervised-learning/boosted_trees_regression.html). Accessed: 2020- 05-19.

[20] S. N. Boyapati and R. Mummidi, ‘Predicting sales using Machine Learning Techniques’, Dissertation, 2020.

[21] S. Cheriyan, S. Ibrahim, S. Mohanan and S. Treesa, "Intelligent Sales Prediction Using Machine Learning Techniques," 2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE), 2018, pp. 53-58, doi: 10.1109/iCCECOME.2018.8659115.

[22] A. Krishna, A. V, A. Aich and C. Hegde, "Sales-forecasting of Retail Stores using Machine Learning Techniques," 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS), 2018, pp. 160-166, doi: 10.1109/CSITSS.2018.8768765.

[23] R. P and S. M, "Predictive Analysis for Big Mart Sales Using Machine Learning Algorithms," 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021, pp. 1416-1421, doi: 10.1109/ICICCS51141.2021.9432109.

[24] JN Hu, JJ Hu, HB Lin, XP Li, CL Jiang, XH Qiu, and WS Li. State-of-charge estimation for battery management system using optimized support vector ma- chine for regression. *Journal of Power Sources*, 269:682–693, 2014.

[25] Wangchao Lou, Xiaoqing Wang, Fan Chen, Yixiao Chen, Bo Jiang, and Hua Zhang. Sequence based prediction of dna-binding proteins based on hybrid feature selection using random forest and gaussian naive bayes. *PloS one*, 9(1), 2014.

DISSEMINATION OF WORK

PUBLICATION DETAILS

SR. NO.	PAPER TITLE	CONFERENCE NAME	CONFERENCE DURATION	ISBN NUMBER
1	Deploying Machine Learning Model on Cloud for Supermarket Sales Data Analytics and Forecasting	JOURNAL OF INFORMATION AND COMPUTATIONAL SCIENCE (Volume 12 Issue 5, May 2023)	March 26, 2023	ISSN: 1548-7741

Journal of Information and Computational Science

JOURNAL OF INFORMATION AND COMPUTATIONAL SCIENCE

ACCEPTANCE LETTER TO AUTHOR

Dear Author,

With reference to your paper submitted **"Deploying Machine Learning Model on Cloud for Supermarket Sales Data Analytics and Forecasting."**

We are pleased to accept the same for publication in **JoICS**.

Manuscript ID: JOICS/6676

Maintenance/processing fee of 2000 INR Per paper. Please note that the amount we are charging is very nominal & only an online maintenance and processing fee.

The Fee includes:
Online maintenance and processing charge
No limitation of number of pages
Editorial fee and Taxes

Note: Fee paid for the publication of the paper does not refund under any circumstances.

In case of any query please do not hesitate to contact us at submitjoics@gmail.com . Early reply is appreciated.

DATE
25-MARCH-2023

Sincerely,
Best regards,
Joseph Sung

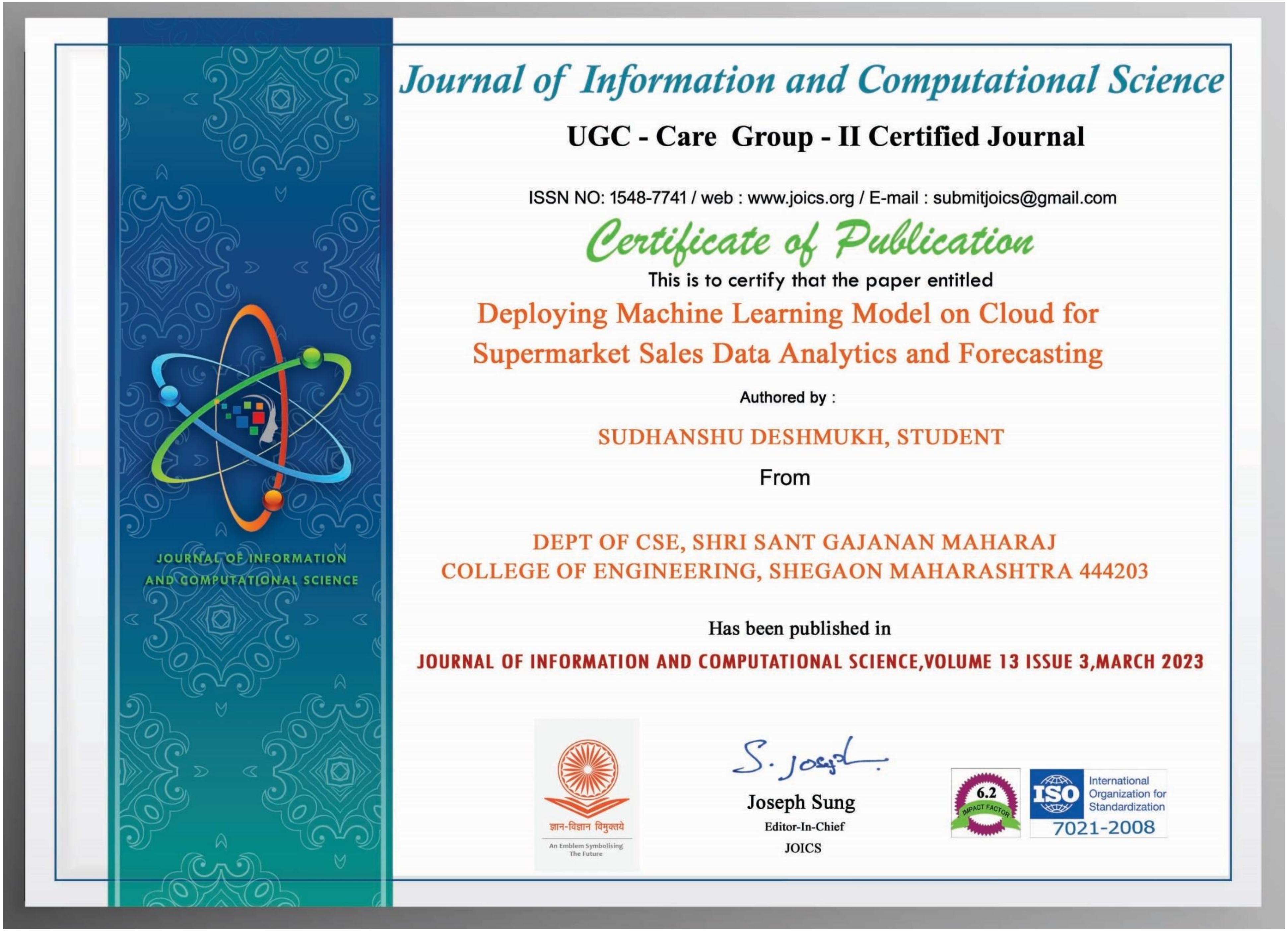
http://www.joics.or

S. Joseph
Editor-In-Chief
Joseph Sung

ISO International Organization for Standardization 7021-2008

ज्ञान-विज्ञान विमुक्तये
An Emblem Symbolising The Future

DOI:10.12733.JICS
crossref member
CROSSREF.ORG
THE CITATION LINKING BACKBONE



JOURNAL OF INFORMATION AND COMPUTATIONAL SCIENCE

Journal of Information and Computational Science

UGC - Care Group - II Certified Journal

ISSN NO: 1548-7741 / web : www.joics.org / E-mail : submitjoics@gmail.com

Certificate of Publication

This is to certify that the paper entitled

Deploying Machine Learning Model on Cloud for Supermarket Sales Data Analytics and Forecasting

Authored by :

SUDHANSHU DESHMUKH, STUDENT

From

DEPT OF CSE, SHRI SANT GAJANAN MAHARAJ
COLLEGE OF ENGINEERING, SHEGAON MAHARASHTRA 444203

Has been published in

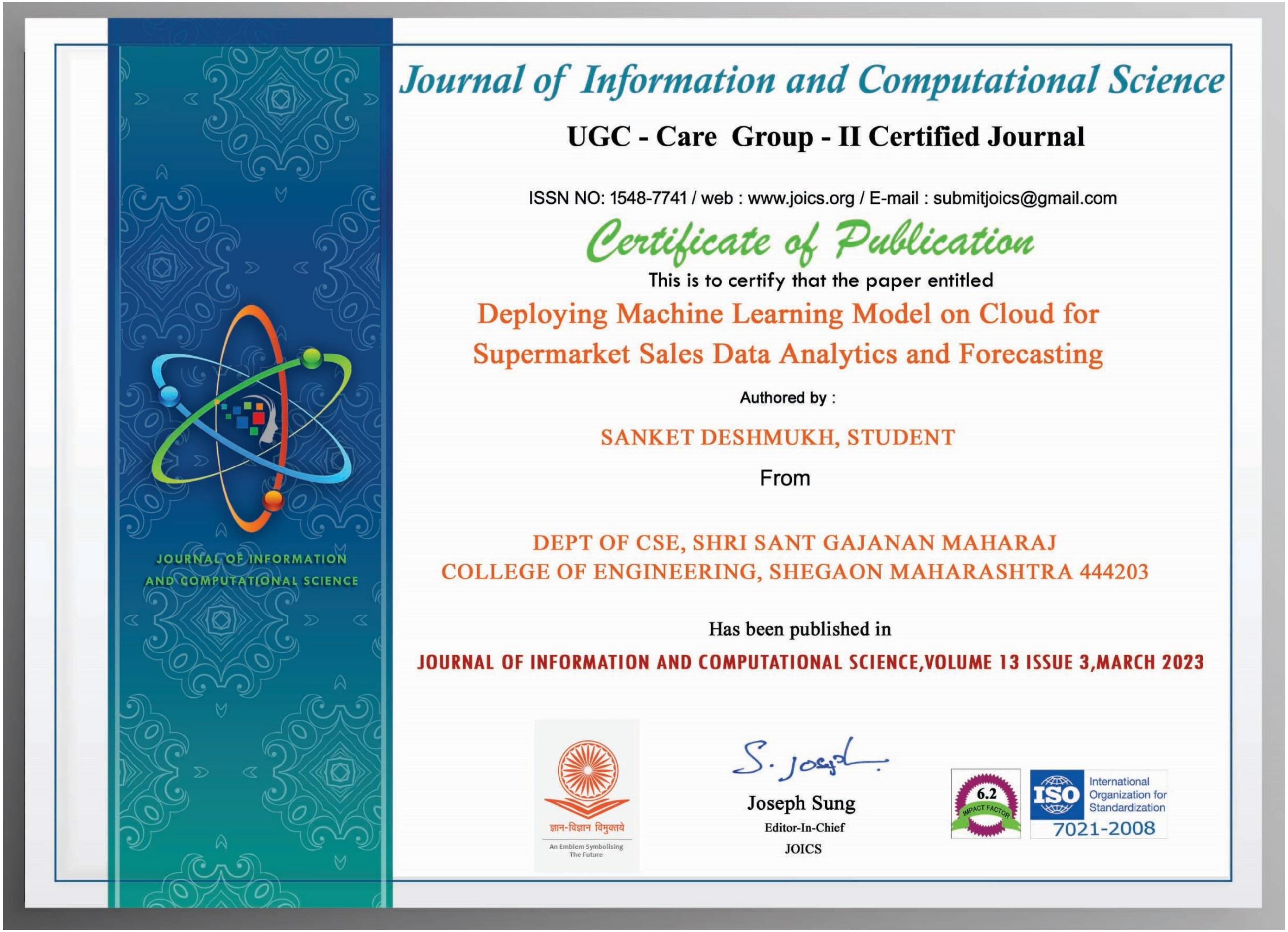
JOURNAL OF INFORMATION AND COMPUTATIONAL SCIENCE,VOLUME 13 ISSUE 3,MARCH 2023

ज्ञान-विज्ञान विमुक्तये
An Emblem Symbolising The Future

S. Joseph
Joseph Sung
Editor-In-Chief
JOICS

6.2 IMPACT FACTOR

ISO International Organization for Standardization 7021-2008



JOURNAL OF INFORMATION
AND COMPUTATIONAL SCIENCE

Journal of Information and Computational Science

UGC - Care Group - II Certified Journal

ISSN NO: 1548-7741 / web : www.joics.org / E-mail : submitjoics@gmail.com

Certificate of Publication

This is to certify that the paper entitled

Deploying Machine Learning Model on Cloud for Supermarket Sales Data Analytics and Forecasting

Authored by :

SANKET DESHMUKH, STUDENT

From

DEPT OF CSE, SHRI SANT GAJANAN MAHARAJ
COLLEGE OF ENGINEERING, SHEGAON MAHARASHTRA 444203

Has been published in

JOURNAL OF INFORMATION AND COMPUTATIONAL SCIENCE,VOLUME 13 ISSUE 3,MARCH 2023

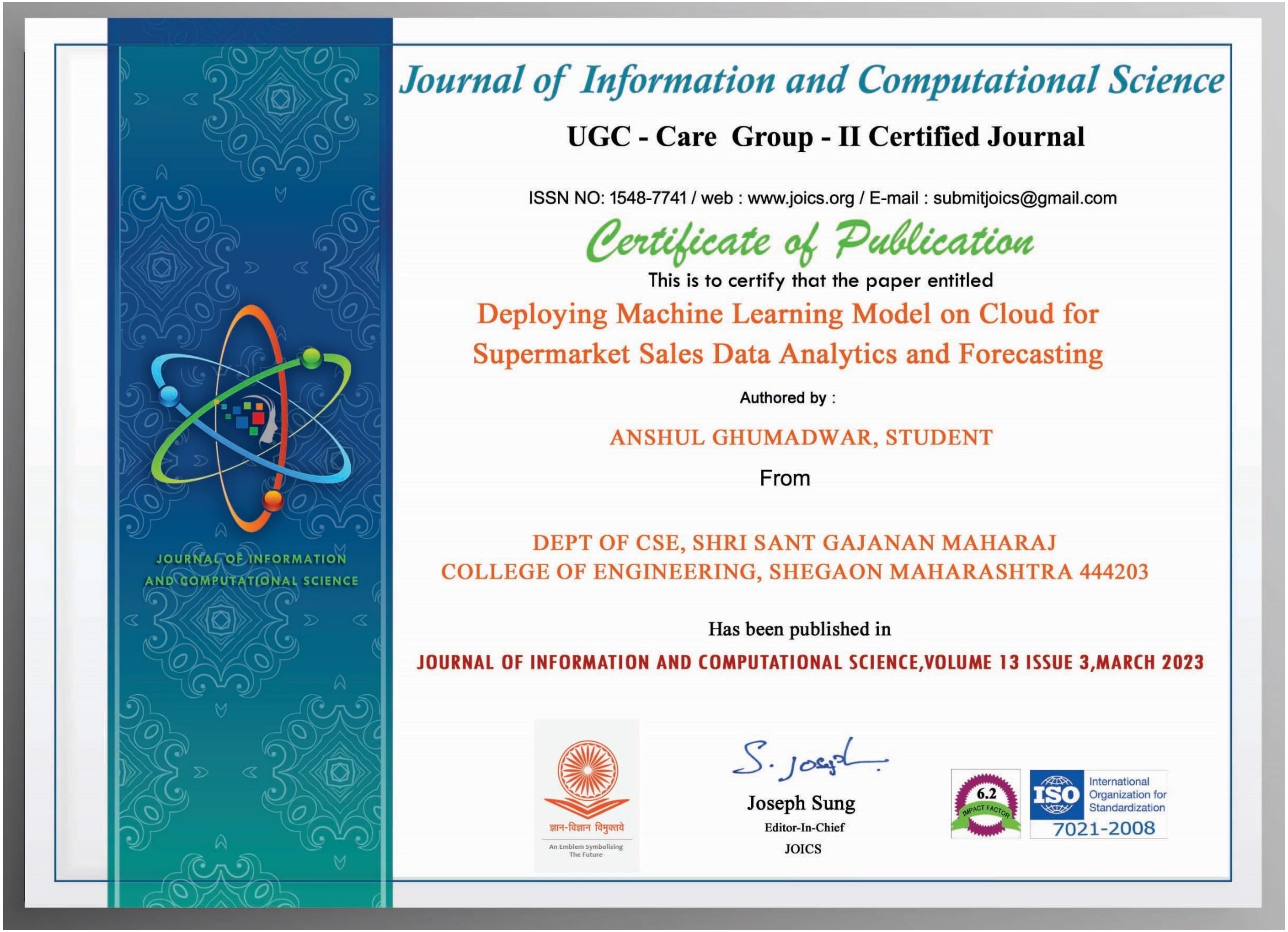
ज्ञान-विज्ञान विमुक्तये
An Emblem Symbolising
The Future

S. Joseph

Joseph Sung
Editor-In-Chief
JOICS

6.2
IMPACT FACTOR

ISO
International
Organization for
Standardization
7021-2008



Journal of Information and Computational Science

UGC - Care Group - II Certified Journal

ISSN NO: 1548-7741 / web : www.joics.org / E-mail : submitjoics@gmail.com

Certificate of Publication

This is to certify that the paper entitled

Deploying Machine Learning Model on Cloud for Supermarket Sales Data Analytics and Forecasting

Authored by :

ANSHUL GHUMADWAR, STUDENT

From

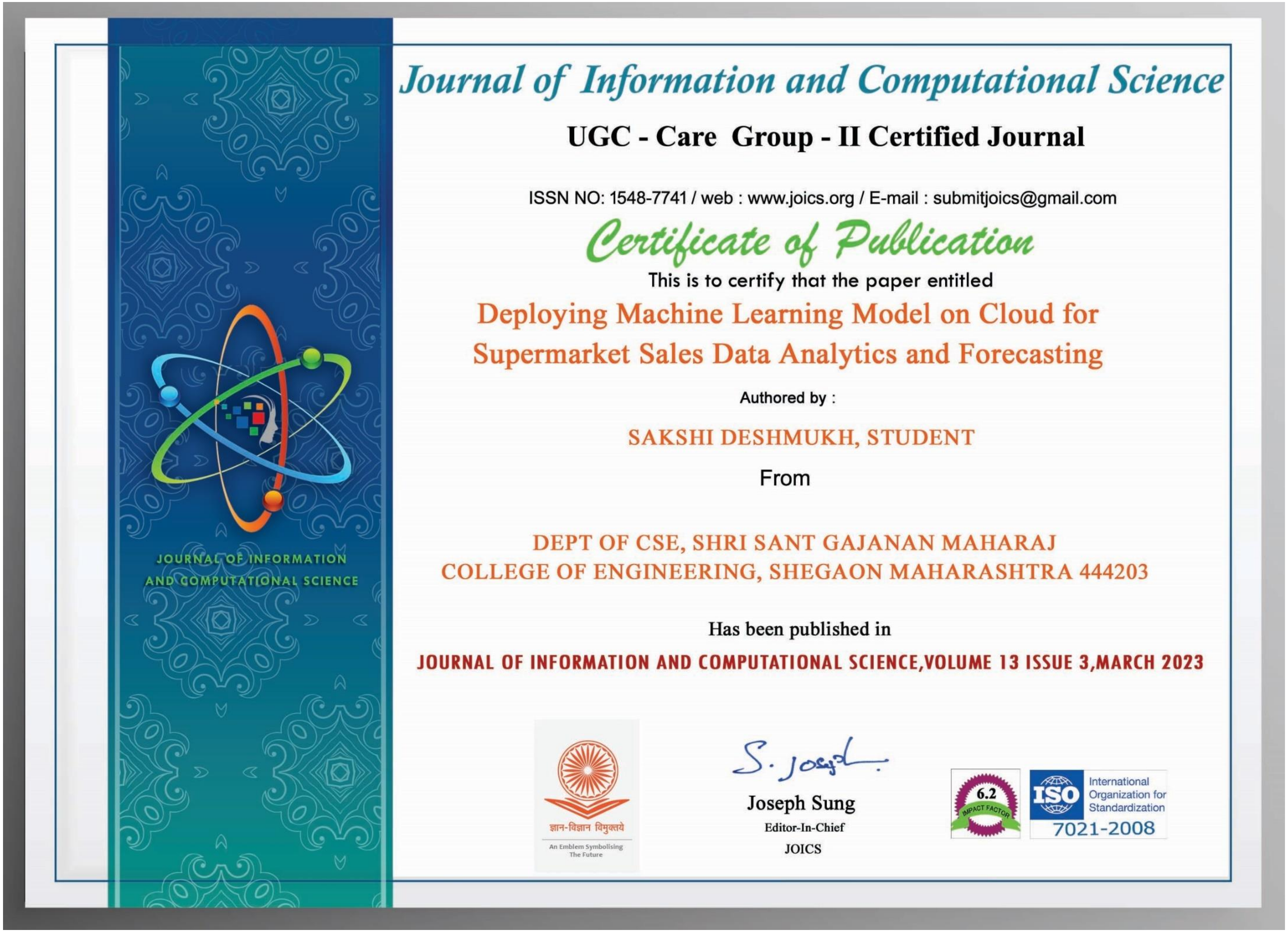
DEPT OF CSE, SHRI SANT GAJANAN MAHARAJ
COLLEGE OF ENGINEERING, SHEGAON MAHARASHTRA 444203

Has been published in

JOURNAL OF INFORMATION AND COMPUTATIONAL SCIENCE,VOLUME 13 ISSUE 3,MARCH 2023

S. Joseph
Joseph Sung
Editor-In-Chief
JOICS

JOURNAL OF INFORMATION
AND COMPUTATIONAL SCIENCE
ज्ञान-विज्ञान विमुक्तये
An Emblem Symbolising The Future
6.2
IMPACT FACTOR
ISO
International Organization for Standardization
7021-2008



JOURNAL OF INFORMATION
AND COMPUTATIONAL SCIENCE

Journal of Information and Computational Science

UGC - Care Group - II Certified Journal

ISSN NO: 1548-7741 / web : www.joics.org / E-mail : submitjoics@gmail.com

Certificate of Publication

This is to certify that the paper entitled

Deploying Machine Learning Model on Cloud for Supermarket Sales Data Analytics and Forecasting

Authored by :

SAKSHI DESHMUKH, STUDENT

From

DEPT OF CSE, SHRI SANT GAJANAN MAHARAJ
COLLEGE OF ENGINEERING, SHEGAON MAHARASHTRA 444203

Has been published in

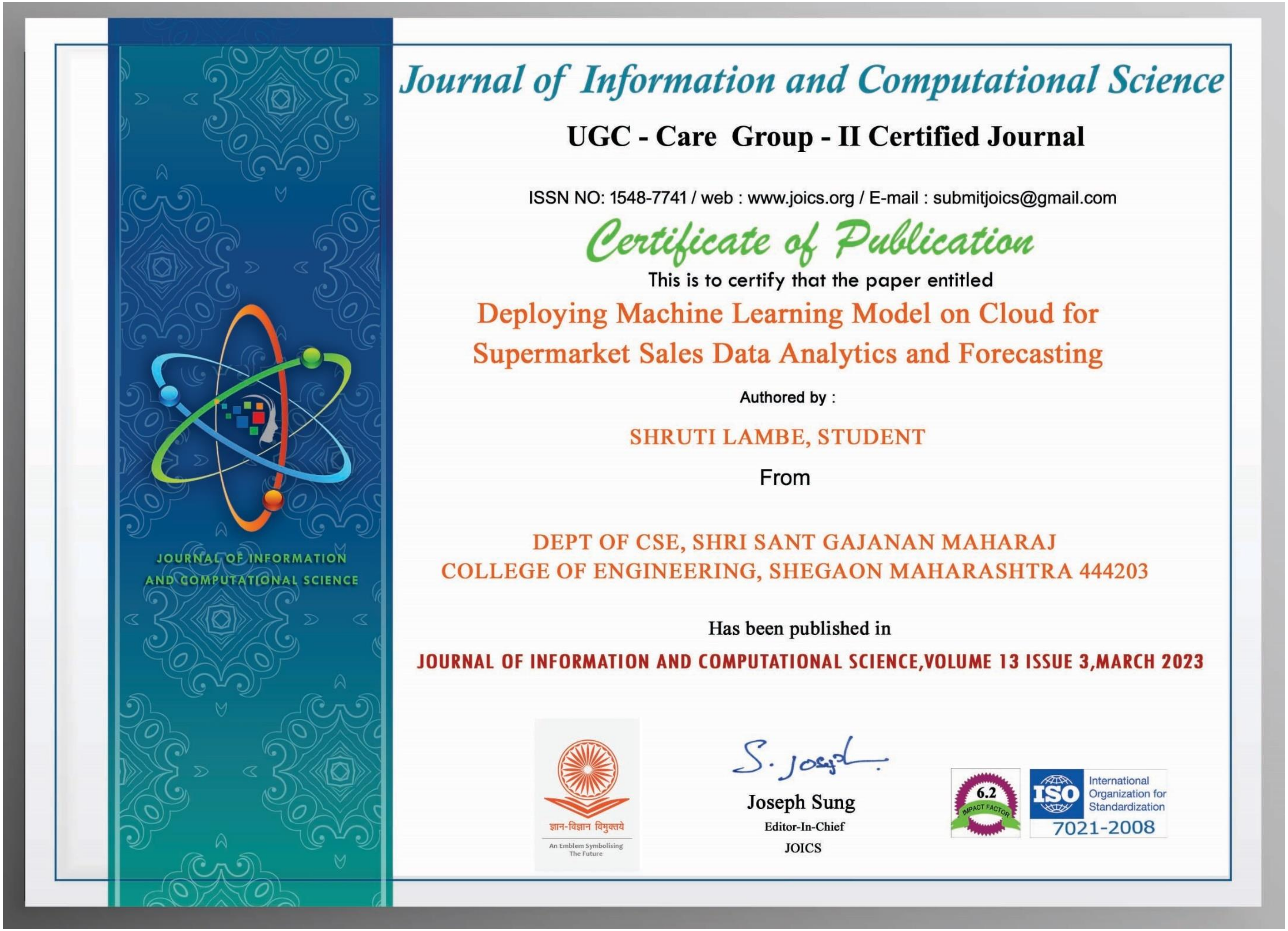
JOURNAL OF INFORMATION AND COMPUTATIONAL SCIENCE,VOLUME 13 ISSUE 3,MARCH 2023

ज्ञान-विज्ञान विमुक्तये
An Emblem Symbolising
The Future

S. Joseph
Joseph Sung
Editor-In-Chief
JOICS

6.2
IMPACT FACTOR

ISO
International
Organization for
Standardization
7021-2008



JOURNAL OF INFORMATION
AND COMPUTATIONAL SCIENCE

Journal of Information and Computational Science

UGC - Care Group - II Certified Journal

ISSN NO: 1548-7741 / web : www.joics.org / E-mail : submitjoics@gmail.com

Certificate of Publication

This is to certify that the paper entitled

Deploying Machine Learning Model on Cloud for
Supermarket Sales Data Analytics and Forecasting

Authored by :

SHRUTI LAMBE, STUDENT

From

DEPT OF CSE, SHRI SANT GAJANAN MAHARAJ
COLLEGE OF ENGINEERING, SHEGAON MAHARASHTRA 444203

Has been published in

JOURNAL OF INFORMATION AND COMPUTATIONAL SCIENCE,VOLUME 13 ISSUE 3,MARCH 2023

ज्ञान-विज्ञान विमुक्तये
An Emblem Symbolising
The Future

Joseph Sung
Editor-In-Chief
JOICS

6.2
IMPACT FACTOR

ISO
International
Organization for
Standardization
7021-2008