BIGMART SALES PREDICTION USING MACHINE LEARNING

Mini Project Report submitted to
Jawaharlal Nehru Technological University Hyderabad
in partial fulfillment for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering

by

ABDUR RAHMAN FAISAL
Roll No:19X31A0502

Under the Guidance of
MRS. M. SRUTHI
ASSOCIATE PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRI INDU INSTITUTE OF ENGINEERING &TECHNOLOGY

(Affiliated to JNTUH, Hyderabad, Approved by AICTE, New Delhi)

Sheriguda (V), Ibrahimpatnam (M), R.R.Dist., Telangana- 501510.

Sri Indu Institute of Engineering & Technology

Grade, Recognized under 2(f) of UGC Act 1956.
(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)
Khalsa Ibrahimpatnam, Sheriguda(V), Ibrahimpatnam(M), Ranga Reddy Dist., Telangana – 501 510
Website : https://siiet.ac.in/

CERTIFICATE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

This is to certify that the report entitled **"BIGMART SALES PREDICTION USING MACHINE LEARNING"**, being submitted by **ABDUR RAHMAN FAISAL,** bearing **Roll No: 19X31A0502,** to **Jawaharlal Nehru Technological University Hyderabad** in partial fulfillment of the requirements for the award of the degree of ***Bachelor of Technology*** in ***Computer Science & Engineering,*** is a record of bonafide work carried out by him. The results of investigations enclosed in this report have been verified and found satisfactory. The results embodied in this report have not been submitted to any other University or Institute for the award of any other degree.

INTERNAL GUIDE **HEAD OF THE DEPARTMENT**

PRINCIPAL **EXTERNAL EXAMINER**

Sri Indu Institute of Engineering & Technology
Grade, Recognized under 2(f) of UGC Act 1956.
(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)
Khalsa Ibrahimpatnam, Sheriguda(V), Ibrahimpatnam(M), Ranga Reddy Dist., Telangana – 501 510
Website : https://siiet.ac.in/

DECLARATION

I,**ABDUR RAHMAN FAISAL** bearing Roll No **19X31A0502,** hereby certify that the dissertation entitled **"BIGMART SALES PREDICTION USING MACHINE LEARNING"**, carried out under the guidance of **MRS M. SRUTHI ,ASSOCIATE PROFESSOR** is submitted to **Jawaharlal Nehru Technological University Hyderabad** in partial fulfillment of the requirements for the award of the degree of ***Bachelor of Technology*** in ***Computer Science & Engineering.*** This is a record of bonafide work carried out by me and the results embodied in this dissertation have not been reproduced or copied from any source. The results embodied in this dissertation have not been submitted to any other University or Institute for the award of any other degree.

Date: **ABDUR RAHMAN FAISAL**

Roll No: **19X31A0502**

Department of CSE, SIIET

CERTIFICATE

This is to certify that Mr. ABDUR RAHMAN FAISAL with **ROLL NO: 19X31A0502** is the bonafide student of **SRI INDU INSTITUTE OF ENGINEERING AND TECHNOLOGY, Hyderabad,** Studying B.Tech (CSE) final year has completed the Major Project entitled "**BigMart Sales Prediction using Machine Learning**" on the Department of Computer Science and Engineering, in partial fulfillment for the award of degree of **Bachelor of Technology in Computer Science and Engineering.**

For Conscience Technologies

Project Manager

ACKNOWLEDGEMENT

With great pleasure I take this opportunity to express my heartfelt gratitude to all the persons who helped me in making this project work a success.

First of all I am highly indebted to **Principal, Dr. I. SATYANARAYANA** for giving me the permission to carry out this project.

I would like to thank **Dr. B.RATNAKANTH** Professor & Head of the Department (CSE), for giving support through out the period of my study in SIIET. I am grateful for his valuable suggestions and guidance during the execution of this project work.

My sincere thanks to project guide **Mrs. M. Sruthi** for potentially explaining the entire system and clarifying the queries at every stage of the project.

My whole hearted thanks to the staff of **SIIET** who co-operated us for the completion of the project in time.

Last but not the least, I express my sincere thanks to **Mr. R. VENKAT RAO**, **Secretary,** Sri Indu Group of Institutions, for his continuous encouragement.

I also thank my parents and friends who aided me in completion of the project.

ABDUR RAHMAN FAISAL
(19X31A0502)

Department of Computer Science and Engineering

INSTITUTE VISION

To become a premier institute of academic excellence by providing the world class education that transforms individuals into high intellectuals, by evolving them as empathetic and responsible citizens through continuous improvement.

INSTITUTE MISSION

IM1 : To offer outcome-based education and enhancement of technical and practical skills.

IM2 : To continuous assess of teaching-learning process through institute-industry collaboration.

IM3 : To be a Centre of excellence for innovative and emerging fields in technology development with state-of-art facilities to faculty and students fraternity.

IM4 : To create an enterprising environment to ensure culture, ethics and social responsibility among the stakeholders.

Department of Computer Science and Engineering

DEPARTMENT VISION

To become a prominent knowledge hub for learners, strive for educational excellence with innovative and industrial techniques so as to meet the global needs.

DEPARTMENT MISSION

DM1 : To provide ambiance that enhances innovations, problem solving skills, leadership qualities, decision making, team-spirit and ethical responsibilities.

DM2 : To impart quality education with professional and personal ethics, so as to meet the challenging technological needs of the industry and society.

DM3 : To provide academic infrastructure and develop linkage with the world class organizations to strengthen industry-academia relationships for learners.

DM4 : To provide and strengthen new concepts of research in the thrust area of Computer Science and Engineering to reach the needs of Government and Society.

ABSTRACT

Everybody wants to know how to buy goods cheaper or how to advertise them at low cost. Here is the answer. That is Big Mart. Big Mart is Online one stop marketplace where you can buy or sell or advertise your merchandise at low cost. The goal is to make Big Mart the shopping paradise for buyers and the marketing solutions for the sellers. The ultimate goal is to prosper with customers. The project "**BIGMART SALES DATASET**" aims to build a predictive model and find out the sales of each product at a particular store. Big Mart will use this model to understand the properties of products and stores which play a key role in increasing sales. This can also be done based on the hypothesis that should be done before looking at the data. Machine Learning is a category of algorithms that allows software applications to become more accurate in predicting outcomes without being explicitly programmed. The basic premise of machine learning is to build models and employ algorithms that can receive input data and use statistical analysis to predict an output while updating outputs as new data becomes available. Here we employ models such as the Linear Regressor, Random Forest Regressor and the Lasso Regressor. These models can be applied in different areas and trained to match the expectations of management so that accurate steps can be taken to achieve the organization's target. In this paper, the case of Big Mart, a one-stop-shopping-center, has been discussed to predict the sales of different types of items and for understanding the effects of different factors on the items' sales. Taking various aspects of a data set collected for Big Mart, and the methodology followed for building a predictive model, results with high levels of accuracy are generated, and these observations can be employed to take decisions to improve sales.

CONTENTS

LIST OF FIGURES

Following are the list of figures used in this project documentation at various locations.

LIST OF SCREENS

Following are the list of Screens developed in this project at various stages.

INTRODUCTION

With the rapid development of global malls and stores chains and the increase in the number of electronic payment customers, the competition among the rival organizations is becoming more serious day by day. Each organization is trying to attract more customers using personalized and short-time offers which makes the prediction of future volume of sales of every item an important asset in the planning and inventory management of every organization, transport service, etc. Due to the cheap availability of computing and storage, it has become possible to use sophisticated machine learning algorithms for this purpose. In this project, we are providing forecast for the sales data of big mart in a number of big mart stores across various location types which is based on the historical data of sales volume.

1.1 Motivation

Sales Forecasting will allow you to set clear goals and expectations for your business. Your ambitions will be based on data, not on numbers pulled from thin air. You'll be able to look at your past success and build from that foundation. Here with the help of the sales prediction, its very easy for the seller to predict the sales of a particular product easily and it also helps the customer to find the product they need at low costs and with greater availability. Here sales forecast can help you with staffing and other anticipated needs. When you know what's coming, you'll be able to have the staff on hand that you need, as well as the inventory to meed the additional volume. You'll be able to allocate resources and prepare for market changes. As good sales are the life of every organization so the forecasting of sales plays an important role in any shopping complex. Always a better prediction is helpful, to develop as well as to enhance the strategies of business about the marketplace which is also helpful to improve the knowledge of marketplace.

1.2 Problem definition

Day by day competition among different shopping malls as well as big marts is getting more serious and aggressive only due to the rapid growth of the global malls and on-line shopping. Every mall or mart is trying to provide personalized and

short-time offers for attracting more customers depending upon the day, such that the volume of sales for each item can be predicted for inventory management of the organization, logistics and transport service, etc. Here in today's world, the predictions are done on pure assumptions without prior calculations which results in low sales of the company. Here also the predictions are done on inaccurate data by which the predictions are wrong. And these systems were very time-consuming and costly when compared to machine learning algorithm which are very sophisticated and provide techniques to predict or forecast the future demand of sales for an organization. Different machine learning algorithms like linear regression analysis, random forest, etc are used for prediction or forecasting of sales volume

1.3 Objective of Project

The objective is to create a model that can predict the sales per product for each store. Using this model, BigMart will try to understand the properties of products and stores which play a key role in increasing sales. Here objective is also to make BigMart, a shopping paradise for the customer and the best marketing place and selling place for the seller.

1.4 Limitations of the Project

Here in this project “Bigmart Sales Prediction”, there is a further need of experiments for proper measurements of both accuracy and resource efficiency to assess and optimize correctly. Moreover, the flexibility of the proposed approach can be increased with variants at a very appropriate stage of regression model-building. Here also, various stakeholders concerned with sales information can also provide more inputs to help in hypothesis generation and more instances can be taken into consideration such that more precise results that are closer to real world situations are generated.

LITERATURE SURVEY

2.1 Introduction

The method for long term electric power forecasting using long term annual growth factors was proposed [1]. Prediction and analysis of aero-material consumption based on multivariate linear regression model was proposed by collecting the data of basic monitoring indicators of aircraft tire consumption from 2001 to 2016 [2].Sales forecasts provide insight into how a firm should manage its workforce, cash flow, and the means. This is an important precondition for the planning and decision-making of enterprises. It allows businesses to formulate their business plans effectively. Learning algorithms used in classification and model categories such as linear Regression, Ridge Regression, Random Forest, Decision Tree etc.

Makridakis,S., Wheelwrigh.S.C., Hyndman. R.J Forecasting methods and applications (2008) Forecasting methods and applications contain a Lack of Data and short life cycles. So, some of the data like historical data, consumer- oriented markets face uncertain demands, can be a prediction for an accurate result.

In 2012 O. Ajao Isaac, A. Abdullahi Adedeji, I. Raji Ismail Regression analysis is used across business fields for tasks as diverse as systematic risk estimation, production and operations management, and statistical inference. The study reveals that polynomial regression is a better alternative with a very high coefficient of determination.

In 2013 X. Yua, Z. Qi, Y. Zhao Advances in information technologies have changed our lives in many ways. There is a trend that people look for news and stories on the internet. Previously regression models can suffer from the over-fitting problem. Recent theoretical studies in statistics proposed a novel method, namely support vector regression (SVR), to overcome the over-fitting problem.

In 2015 Xinqing Shu, Pan Wang Boosting is one of the algorithms which can boost the accuracy of weak classifiers, and Adaboost has been widely and successfully applied to classification, detection, and data mining problems. In this paper, a new

method of calculating parameters, Adaboost-AC, which uses the accelerated good fitness function to acquire the weights of the weak classifiers is presented. The new algorithm is compared with the traditional Adaboost based on the UCI database and its promising performance is shown by the experimental results.

Das, P., Chaudhury Prediction of retail sales of footwear using feedforward and recurrent Neural Networks (2018) Prediction of retail sales of footwear using feedforward and recurrent neural networks used neural networks for prediction of sales. Using the neural network for predicting weekly retail sales, which is not efficient, So XG boost can work efficiently.

2.2 Existing System

Predicting the sales of each product is not a simple task. Prediction of sales depends on a number factors such as product demand, product supply and many other things. So seller has to himself study all of these things and himself predict the sales. Unfortunately, information about the sales are not always accurate and based on that the seller wrongly predicts the sales and thereby suffers. Existing System includes a process where a seller decides to keep the product based on inaccurate data and thereby, the consumer also suffers with the seller. And the existing system is also slow as it is not automated and the prediction of sales is to be done by the seller himself.

2.2.1 Drawbacks in Existing System

Here in the existing system, when a prediction is done, the results is not accurate which results in loss in the sales. The existing system is also not efficient and time taking as the seller has to himself study all the data of the sales and he has to himself predict the sales because the existing system is not at all automated also. Because of all this problems, its very difficult and impractical for the seller to do all the predictions without causing errors. In the existing system, when the seller is doing the predictions of the sales, it is very prone to errors also. In existing system, there is no management also. Here the seller has to hire an additional man which calculates all this data and predict the sales. Because of this, it incurs extra loss to the company.

2.3 Proposed System

For building a model to predict accurate results the data-set of Big Mart sales undergoes several sequence of steps and in this work we propose a model Bigmart sales prediction using machine learning. Every step plays a vital role for building the proposed model. Here we employ the predicting models such as the linear regressor, random forest regressor and the lasso regressor. By the help of which we can easily predict the sales of each product from various stores with high accuracy. Here all the process is almost automated and the seller has to do nothing by which the sales can increase enormously. Using this model, Big Mart will try to understand the properties of products and stores which play a key role in increasing sales. The aim is to build a predictive model and find out the sales of each product at a particular store. To use various technologies for model building like multiple linear regression analysis and random forest to forecast the sales volume. Various advantages of this system is that it can save a lot of manpower, money and time to the company with increasing the sales to make give a lot of profit to the company.

2.4 Feasibility Study:

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is carried out. This is to ensure that the proposed system is not a burden to the company. For feasibilty analysis, some understanding of the major requirements for the system is essential.

There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operational Feasibility
- Economical Feasibility

Technical Feasibility:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being

placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

BigMart Sales Prediction application is feasible technically, although there is some risk.

The project risk regarding familiarity with Bigmart Sales Prediction system is low.

- The IT Department has strong knowledge of the existing Bigmart Sales Prediction System.

The project risk regarding familiarity with the programming language is low

- The IT department already has rich experience in python programs through web daily maintenance.
- Development tools for system improvement are available on the internet.

The project size is considered medium risk.

- The project team will last for about 3 to 4 months which is from planning to implementation.
- The project team will include 4 people. 3 of them are responsible for web design and maintenance, and 1 person will act as the external consultant.
- The project needs an additional testing team to do system testing and feedback collection work.

Economic feasibility:

Major portion of the cost in this project is for making the application. This system can be easily installed on any computer system and once installed can generate more profit for the company.

Intangible Costs and Benefits :

- Higher sales of the company
- Consumers gets their product always in stock

Social Feasibility:

The aspect of study is to check the level of acceptance of the seller and the consumer . This includes the process of training the user to the system efficiently. A social feasibility study explores the impact of a project on society and of society on the project. For example, this type of study might look at how ambient social structure in the area will affect the number of qualified employees that may be available, or the compatibility of local residents with the project.

2.5 Features of the Project

Bigmart sales prediction is an excellent application for predicting the sales with high accuracy and can employ various predicting algorithms such as the linear regressor, random forest regressor and the lasso regressor by which we can easily compare the algorithms and can further get more accurate results. Here this application can handle lots and lots of data provided and can also replace the incorrect data and the missing data with the mean and mode of the data. The prediction of the sales of the products is done within no time which makes this application more feasible.

2.6 Technologies required for implementation

Python:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

The Python interpreter and the extensive standard library are available in source(open-source) or binary form without charge for all major platforms, and can be freely distributed. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms object-oriented, imperative, functional and procedural, and has a large number of comprehensive standard library.

Keras:

Keras is an open-source high-level Neural Network Library, which is written in Python is capable enough to run on Theano, TensorFlow, or CNTK. It cannot handle low-level computations, so it makes use of the Backend library to resolve it. The Backend library act as a high-level API wrapper for the low-level API, which lets it run on the TensorFLow, CNTK, or Theano. Keras can be developed in R as well as Python, suct that the code can be run with TensorFlow, Theano, CNTK, or MXNet as per the requirements. Keras can run on CPU, NVIDEA, AMD GPU, TPU, etc.

It ensures that producing models with Keras is really simple as it totally supports to run with TensorFlow serving, GPU acceleration(WebKeras, Keras.js), Android(TF,TF Lite), iOS(Native CodeML) and Raspberry Pi.

Tensor Flow:

Tensorflow is a library that is used in machine learning and it is an open-source library for numerical computations. It is used for developing machine learning applications and this library was first created by the Google brain team and it is the most common and successfully used library that provides various tools for machine learning applications. TensorFlow library is used in many companies in the industries like Airbnb. This company applies machine learning using TensorFlow to detect objects and classify the

Numpy: NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

Pandas:

Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy. Pandas makes it simple to do many of the time consuming, repetitive tasks associated with working with data, including: Data cleansing, Data fill, Data normalization, Merges and joins, Data visualization, Statistical analysis, Data inspection, Loading and saving data and much more.

Matplotlib:

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open source alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications.

ANALYSIS

3.1 Introduction

Software analysis and design includes all activities, which help the transformation of requirement specification into implementation. Requirement specifications specify all functional and non-functional expectations from the software. These requirement specifications come in the shape of human readable and understandable documents, to which a computer has nothing to do.

Software analysis and design is the intermediate stage, which helps human-readable requirements to be transformed into actual code.

3.1 Requirement Specification

Requirement Specification plays an important role to create quality software solution. Requirements are refined and analyzed to assess the clarity. Requirements are represented in a manner that ultimately leads to successful software implementation. The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigation from one screen to the other well ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

3.2.1 User Requirements

User requirements are statements in natural language along with corresponding diagrams (tables, forms, intuitive diagrams) detailing the services provided by the system and operational constraints it must comply with. Additionally, it’s worth noting that user requirements primarily focus on the user’s needs. Thus, these user requirements cater to the customer.

Essentially, it entails the requirement that the user wants or ability to perform a functionality or action with the system. So, it outlines the activities a user can perform with the system. User requirements entail both functional and non-functional requirements that are understandable even by users without technical knowledge.

3.2.1.1 Functional Requirements

- Login: Here the store managers can login with their accounts
- Provide data: The store managers can easily provide the Bigmart Data-set.

3.2.1.1 Non-Functional Requirements

- here the application can handle large data-set.
- Server running online 24*7 without problems.
- The bigmart data-set should be highly secured.
- The application must be portable, scalable, precise and legal.

3.2.1 Hardware Requirements

SYSTEM	:	Pentium IV 2.4GHz
HARD DISK	:	20GHZ
FLOPPY DRIVE	:	1.44Mb
MONITOR	:	14’ Colour Monitor
MOUSE	:	Optical Mouse
RAM	:	512Mb

3.2.1 Software Requirements

OPERATING SYSTEM	:	Windows 7 Ultimate
CODING LANGUAGE	:	Python
FRONT-END	:	Python
DESIGNING	:	HTML, CSS, JAVASCRIPT
DATABASE	:	MySQL

DESIGN

4.1 Introduction

Software design is a mechanism to transform user requirements into some suitable form, which helps the programmer in software coding and implementation. It deals with representing the client's requirement, as described in SRS (Software Requirement Specification) document, into a form, i.e., easily implementable using programming language.

The software design phase is the first step in SDLC (Software Design Life Cycle), which moves the concentration from the problem domain to the solution domain. In software design, we consider the system to be a set of components or modules with clearly defined behaviors & boundaries.

During design, progressive refinement of data structure, program structure and procedural details are developed, reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design, procedural design.

4.2 Proposed System Architecture

The architecture of a system reflects how the system is used and how it interacts with other systems and the outside world. It describes the interconnection of all the system’s components and the data link between them. The architecture of a system reflects the way it is thought about in terms of its structure, functions, and relationships. In architecture, the term “system” usually refers to the architecture of the software itself, rather than the physical structure of the buildings or machinery. The architecture of a system reflects the way it is used, and therefore changes as the system is used.

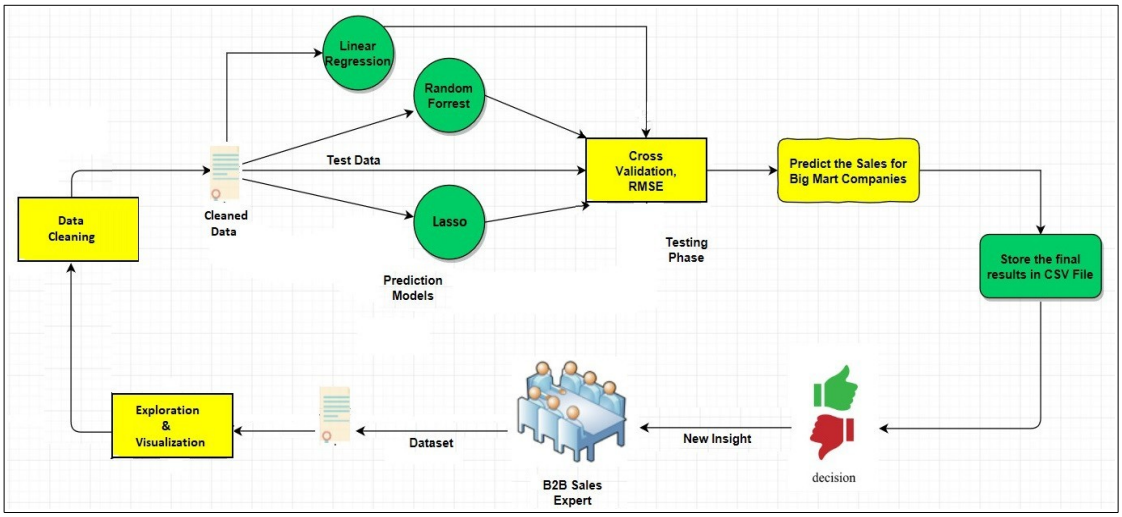


Fig 4.2: Architecture model of BigMart Sales Prediction Apllication

Here from the above figure, we can clearly see the architecture where when the dataset is provided by the Sales Expert, the data exploration and visualizations begins.

Here the term **data exploration** means in this phase useful information about the data has been extracted from the dataset i.e., trying to identify the information from hypotheses vs available data.

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. Additionally, it provides an excellent way for employees or business owners to present data to non-technical audiences without confusion.

Univariate Plots

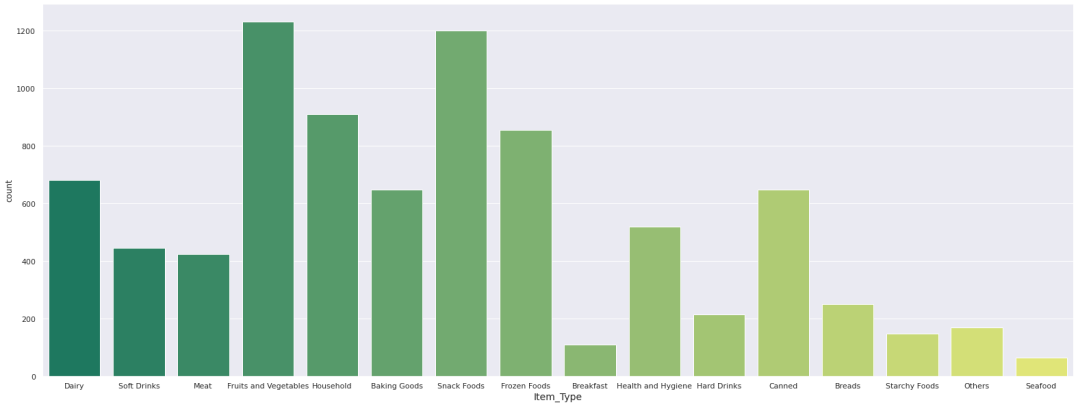


Fig 4.2.1: Uni-variate Plots

Multivariate Plots

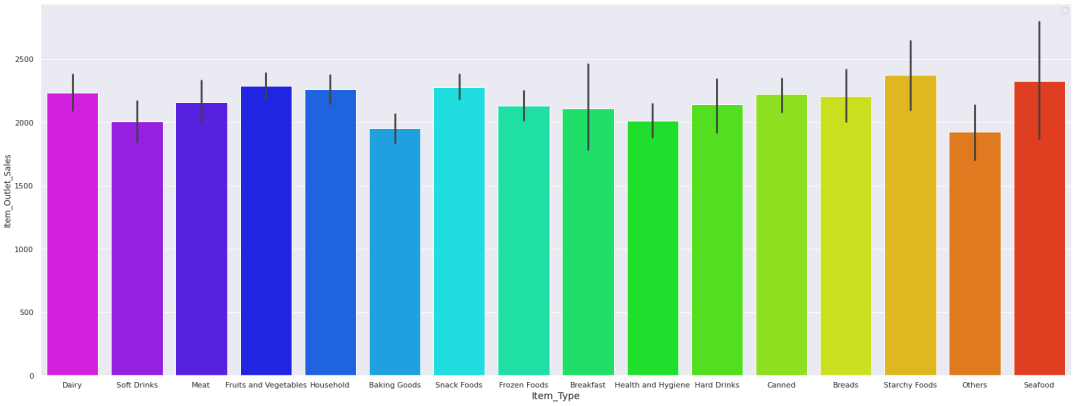


Fig 4.2.2: Multi-variate Plots

Corelation Matrix

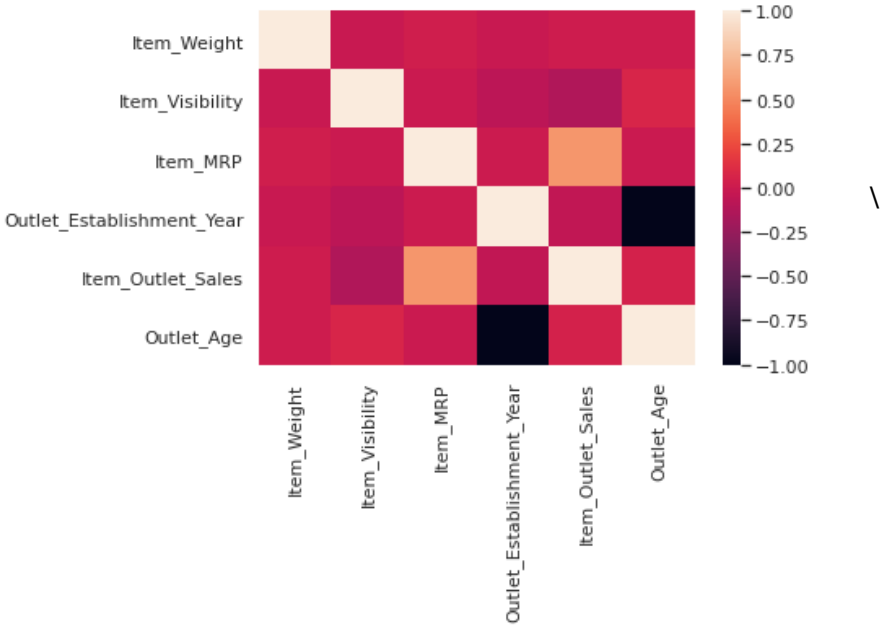


Fig 4.2.3: Corelation Matrix

Data-Cleaning is used when in our dataset, some data has missing values, then the missing data is replaced with the mean and mode which diminishes the correlation among imputed attributes.

After that the cleaned data is taken and on it the predicting models are used which are the linear regressor, the random forest regressor and the lasso regressor.

After that the result is predicted and we can easily compare the predicted results of these models and get the data with atmost accuracy.

4.3 UML DIAGRAMS

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

Mainly, UML has been used as a general-purpose modeling language in the field of software engineering. However, it has now found its way into the documentation of several business processes or workflows. For example, activity diagrams, a type of UML diagram, can be used as a replacement for flowcharts. They provide both a more standardized way of modeling workflows as well as a wider range of features to improve readability and efficacy.

4.3.1 Use Case Diagram

Use case diagrams are a set of use cases, actors, and their relationships. They represent the use case view of a system.

A use case represents a particular functionality of a system. Hence, use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors.

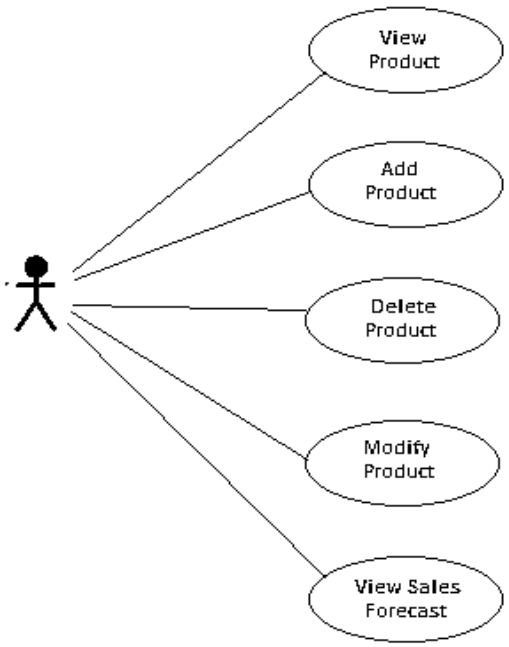


Fig 4.3.1: Use-Case Diagram

4.3.2 Class Diagram

Class diagrams are the most common diagrams used in UML. Class diagram consists of classes, interfaces, associations, and collaboration. Class diagrams basically represent the object-oriented view of a system, which is static in nature. Active class is used in a class diagram to represent the concurrency of the system. Class diagram represents the object orientation of a system. Hence, it is generally used for development purpose. This is the most widely used diagram at the time of system construction.

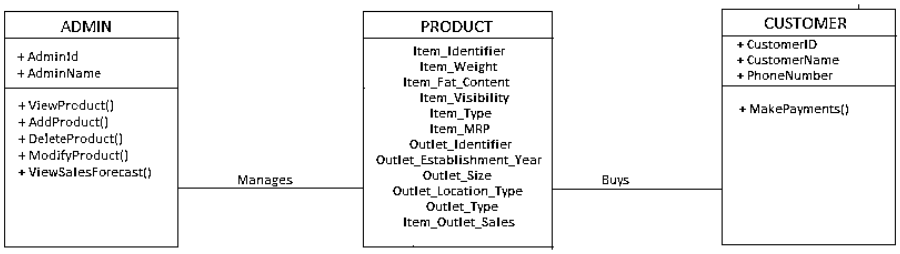


Fig4.3.2: Class Diagram

4.3.3 Sequence Diagram

A sequence diagram is an interaction diagram. From the name, it is clear that the diagram deals with some sequences, which are the sequence of messages flowing from one object to another.

Interaction among the components of a system is very important from implementation and execution perspective. Sequence diagram is used to visualize the sequence of calls in a system to perform a specific functionality.

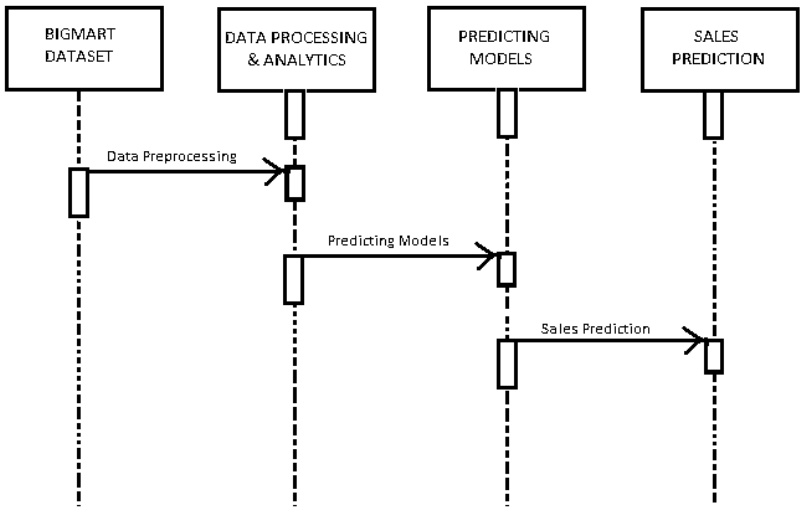


Fig4.3.3: Sequence Diagram

4.3.4 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unifiedd Modelling language(UML), activity diagrams are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores.

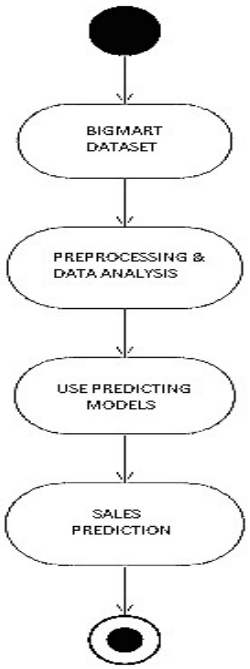


Fig 4.3.4: Activity Diagram

4.3.5 Component Diagram

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node.

It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component is a single unit of the system, which is replaceable and executable. The implementation details of a component are hidden, and it necessitates an interface to execute a function. It is like a black box whose behavior is explained by the provided and required interfaces.

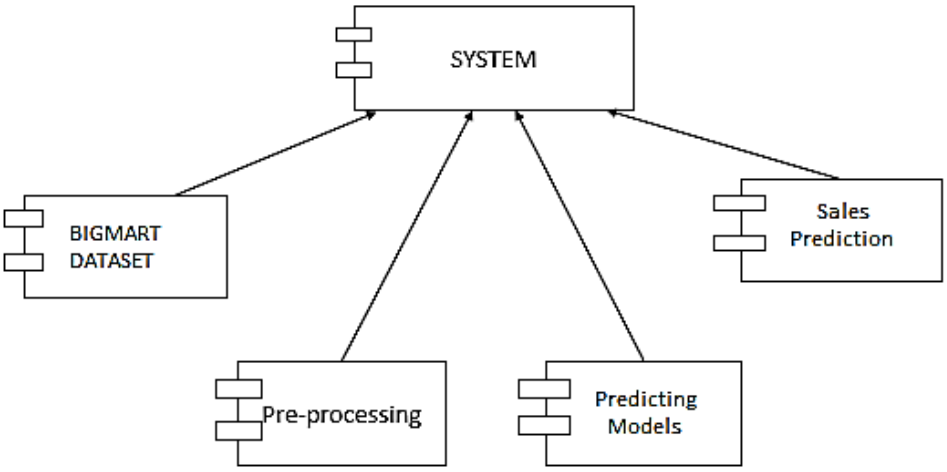


Fig 4.3.5: Component Diagram

4.4 Module design and organization

Here, in our project, various modules that are used are the Bigmart Dataset, Preprocessing and the Predicting models. Here the data is taken from various stores. After that the data is given to preprocessing which includes the data exploration, data visualization and the data cleaning. After that, the predicting models i.e., Linear Regressor, Random Forest Regressor and the Lasso Regressor are applied to the dataset.

IMPLEMENTATION

5.1 Introduction

Here we implement the project “bigmart Sales Prediction using machine learning” using python, the programming language which is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. In addition to this, we use frameworks such as numpy, pandas, scikit, matplotlib, opencv-python, keras and tensorflow. The frameworks such as the numpy, pandas and the matplotlib are used for the data visualization purposes. Keras is the framework tightly integrated with the tensorflow which is used to used to build machine learning models. Keras models offer a simple, user friendly way to define a neural network, which will then be built for you ny the tensorflow. Tensorflow is an open-source set of libraries for creting and working with neural networks, such as those in the machine learning and the deep learning. Keras on the other hand is a high-level API that runs on top of tensorflow. Keras simplifies the implementation of complex neural networks with its easy to use framework.

Now to implement the project, we use data predicting models such as the linear regressor, random forest regressor and the lasso regressor. We also install the IDE and the python distribution known as the Anaconda software which helps you create an environment for many different versions of Python and package versions. Anaconda is also used to install, remove, and upgrade packages in your project environments. Furthermore, you may use Anaconda to deploy any required project with a few mouse clicks.

5.2 Key Functions

Models we will use:

- Linear Regression
- Random Forest Regressor
- Lasso Regressor

Linear Regressor: Linear Regression is usually the first algorithm that people learn for Machine Learning and Data Science. Linear Regression is a linear model that assumes a linear relationship between the input variables (X) and the single output variable (y). In general, there are two cases: Single Variable Linear Regression: it models the relationship between a single input variable (single feature variable) and a single output variable. Multi-Variable Linear Regression: (also known as Multivariate Linear Regression), it models the relationship between multiple input variables (multiple features variables) and a single output variable.

Random Forest Regressor: Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model. Random Forest Regression is very similar to Decision Tree Regression, Basically. it's a meta estimator that fits a number of Decision Trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. A Random Forest regressor may or may not perform better than the Decision Tree in regression (while it usually performs better in classification), because of the delicate overfitting-underfitting tradeoff in the nature of tree-constructing algorithms.

Lasso Regressor: LASSO regression is a variation of Linear Regression that uses Shrinkage. Shrinkage is a process that data values are shrunk towards a central point as the mean. This type of regression is well-suited for models showing heavy multicollinearity (heavy correlation of features with each other) or when you want to automate certain parts of model selection, like variable selection/parameter elimination.. The word "LASSO" stands for Least Absolute Shrinkage and Selection Operator. It is a statistical formula for the regularisation of data models and feature selection. The LASSO regression in regularization is based on simple models that posses fewer parameters. Regularization resolves the overfitting problem, which affects the accuracy level of the model. Regularization is executed by the addition of the "penalty" term to the best-fit equation produced by the trained data.

5.3 Sample Code

#IMPORTING PACKAGES

```
import os #paths to file
import numpy as np # linear algebra
import pandas as pd # data processing
import warnings# warning filter

#ploting libraries
import matplotlib.pyplot as plt
import seaborn as sns

#feature engineering
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import LabelEncoder

#train test split
from sklearn.model_selection import train_test_split

#metrics
from sklearn.metrics import mean_absolute_error as MAE
from sklearn.metrics import mean_squared_error as MSE
from sklearn.metrics import r2_score as R2
from sklearn.model_selection  import cross_val_score as CVS

#ML models
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import Lasso

#default theme and settings
sns.set(context='notebook', style='darkgrid', palette='deep', font='sans-serif',
font_scale=1, color_codes=False, rc=None)
pd.options.display.max_columns
```

```
#warning hadle
warnings.filterwarnings("always")
warnings.filterwarnings("ignore")
#FILES PATHS
#list all files under the input directory
for dirname, _, filenames in os.walk('/kaggle/input'):
  for filename in filenames:
      print(os.path.join(dirname, filename))
#path for the training set
tr_path = "/kaggle/input/bigmart-sales-data/Train.csv"
#path for the testing set
te_path = "/kaggle/input/bigmart-sales-data/Test.csv"
#preprocessing of the training dataset-------------------------
#column information
tr_df.info(verbose=True, null_counts=True)
#summary statistics test
te_df.describe()
#summary statistics train
te_df.describe()

#MISSING VALUES
print("test mode, train mode\n",[tr_df['Outlet_Size'].mode().values[0],
te_df['Outlet_Size'].mode().values[0]])
#train
tr_df['Outlet_Size'] = tr_df['Outlet_Size'].fillna(
tr_df['Outlet_Size'].dropna().mode().values[0])
#test
te_df['Outlet_Size'] = te_df['Outlet_Size'].fillna(
te_df['Outlet_Size'].dropna().mode().values[0])
#checking if we filled missing values
tr_df['Outlet_Size'].isnull().sum(),te_df['Outlet_Size'].isnull().sum()
```

#DATA EXPLORATION

```
#SPLIT DATA TO CATEGORICAL AND NUMERICAL DATA
#list of all the numeric columns
num = tr_df.select_dtypes('number').columns.to_list()
#list of all the categoric columns
cat = tr_df.select_dtypes('object').columns.to_list()
#numeric df
BM_num =  tr_df[num]
#categoric df
BM_cat = tr_df[cat]
#print(num)
#print(cat)
[tr_df[category].value_counts() for category in cat[1:]]

#CORRECT THE REPEATING VALUES
#train
tr_df['Item_Fat_Content'].replace(['LF', 'low fat', 'reg'],
                          ['Low Fat','Low Fat','Regular'],inplace = True)
#test
te_df['Item_Fat_Content'].replace(['LF', 'low fat', 'reg'],
                          ['Low Fat','Low Fat','Regular'],inplace = True)

#check result
tr_df.Item_Fat_Content.value_counts()
#creating our new column for both datasets
tr_df['Outlet_Age'], te_df['Outlet_Age']=
tr_df['Outlet_Establishment_Year'].apply(lambda year: 2020 - year),
te_df['Outlet_Establishment_Year'].apply(lambda year: 2020 - year)

##uncomment to check result
#tr_df['Outlet_Age'].head
#te_df['Outlet_Age'].head
```

#DATA VISUALIZATION

#UNIVARIATE PLOTS

#categorical columns:

['Item_Identifier', 'Item_Fat_Content', 'Item_Type', 'Outlet_Identifier',
'Outlet_Size', 'Outlet_Location_Type', 'Outlet_Type']

```
plt.figure(figsize=(6,4))
sns.countplot(x='Item_Fat_Content' , data=tr_df ,palette='mako')
plt.xlabel('Item_Fat_Content', fontsize=14)
plt.show()

plt.figure(figsize=(27,10))
sns.countplot(x='Item_Type' , data=tr_df ,palette='summer')
plt.xlabel('Item_Type', fontsize=14)
plt.show()

plt.figure(figsize=(15,4))
sns.countplot(x='Outlet_Identifier' , data=tr_df ,palette='winter')
plt.xlabel('Outlet_Identifier', fontsize=14)
plt.show()

plt.figure(figsize=(10,4))
sns.countplot(x='Outlet_Size' , data=tr_df ,palette='autumn')
plt.xlabel('Outlet_Size', fontsize=14)
plt.show()

plt.figure(figsize=(10,4))
sns.countplot(x='Outlet_Location_Type' , data=tr_df ,palette='twilight_shifted')
plt.xlabel('Outlet_Location_Type', fontsize=14)
plt.show()
```

Now for the numerical columns:

#list of all the numeric columns

```
num = tr_df.select_dtypes('number').columns.to_list()
#numeric df
BM_num =  tr_df[num]

plt.hist(tr_df['Outlet_Age'])
plt.title("Outlet_Age")
plt.show()

#MULTIVARIATE PLOTS
plt.figure(figsize=(27,10))
sns.barplot('Item_Type' ,'Item_Outlet_Sales', data=tr_df ,palette='gist_rainbow_r')
plt.xlabel('Item_Type', fontsize=14)
plt.legend()
plt.show()
plt.figure(figsize=(27,10))

sns.barplot('Outlet_Identifier' ,'Item_Outlet_Sales',
data=tr_df ,palette='gist_rainbow')
plt.xlabel('Outlet_Identifier', fontsize=14)
plt.legend()
plt.show()

#CORRALTION MATRIX
#plotting the correlation matrix
sns.heatmap(tr_df.corr() ,cmap='rocket')

# Dropping irrelevant columns
tr_fe=tr_fe.drop(['Item_Identifier','Outlet_Identifier','Outlet_Establishment_Year','O
utlet_Type','Item_Type'],axis=1)

te_fe =
te_fe.drop(['Item_Identifier','Outlet_Identifier','Outlet_Establishment_Year','Outlet_
Type','Item_Type'],axis=1)
```

#USING REGRESSION MODELS

```
LR = LinearRegression(normalize=True)
#fit
LR.fit(X_train, y_train)
#predict


y_predict = LR.predict(X_test)
#score variables
LR_MAE = round(MAE(y_test, y_predict),2)
LR_MSE = round(MSE(y_test, y_predict),2)
LR_R_2 = round(R2(y_test, y_predict),4)
LR_CS  = round(CVS(LR, X, y, cv=5).mean(),4)
print(f" Mean Absolute Error: {LR_MAE}\n")
print(f" Mean Squared Error: {LR_MSE}\n")
print(f" R^2 Score: {LR_R_2}\n")
cross_val(LR,LinearRegression(),X,y,5)
Linear_Regression=pd.DataFrame({'y_test':y_test,'prediction':y_predict})
Linear_Regression.to_csv("Linear Regression.csv")


RFR= RandomForestRegressor(n_estimators=200,max_depth=5,
min_samples_leaf=100,n_jobs=4,random_state=101)
#fit
RFR.fit(X_train, y_train)
#predict
y_predict = RFR.predict(X_test)
#score variables
RFR_MAE = round(MAE(y_test, y_predict),2)
RFR_MSE = round(MSE(y_test, y_predict),2)
RFR_R_2 = round(R2(y_test, y_predict),4)
RFR_CS  = round(CVS(RFR, X, y, cv=5).mean(),4)
print(f" Mean Absolute Error: {RFR_MAE}\n")
```

```
print(f" Mean Squared Error: {RFR_MSE}\n")
print(f" R^2 Score: {RFR_R_2}\n")
cross_val(RFR,RandomForestRegressor(),X,y,5)
Random_Forest_Regressor=pd.DataFrame({'y_test':y_test,'prediction':y_predict})
Random_Forest_Regressor.to_csv("Random Forest Regressor.csv")

LS = Lasso(alpha = 0.05)
#fit
LS.fit(X_train,y_train)
#predict
y_predict = LS.predict(X_test)
#score variables
LS_MAE = round(MAE(y_test, y_predict),2)
LS_MSE = round(MSE(y_test, y_predict),2)
LS_R_2 = round(R2(y_test, y_predict),4)
LS_CS  = round(CVS(LS, X, y, cv=5).mean(),4)
print(f" Mean Absolute Error: {LS_MAE}\n")
print(f" Mean Squared Error: {LS_MSE}\n")
print(f" R^2 Score: {LS_R_2}\n")
cross_val(LS,Lasso(alpha = 0.05),X,y,5)
Lasso_Regressor=pd.DataFrame({'y_test':y_test,'prediction':y_predict})
Lasso_Regressor.to_csv("Lasso Regressor.csv")

MAE= [LR_MAE,RFR_MAE,LS_MAE]
MSE= [LR_MSE,RFR_MSE,LS_MSE]
R_2= [LR_R_2,RFR_R_2,LS_R_2]
Cross_score= [LR_CS,RFR_CS,LS_CS]
Models = pd.DataFrame({
'models': ["Linear Regression","Random Forest Regressor","Lasso Regressor"],
'MAE': MAE, 'MSE': MSE, 'R^2':R_2, 'Cross Validation Score':Cross_score})
Models.sort_values(by='MAE', ascending=True)
```

5.4 Method of Implementation

Models we will use:

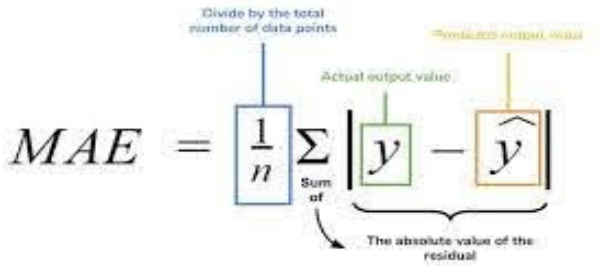
- Linear Regression
- Random Forest Regressor
- Lasso Regressor

The Process of Modeling the Data:

- Importing the model
- Fitting the model
- Predicting Item Outlet Sales
- Regression metrics

Score Metrics for Regression:

- **Mean Absolute Error (MAE)** - Mean of the absolute value of errors (absolute distance from true value): Mean Absolute Error calculates the average difference between the calculated values and actual values. It is also known as scale-dependent accuracy as it calculates error in observations taken on the same scale. It is used as evaluation metrics for regression models in machine learning. It calculates errors between actual values and values predicted by the model. It is used to predict the accuracy of the machine learning model.



$$MAE = \frac{1}{n}\sum \left| y - \hat{y} \right|$$

- **Mean Squared Error (MSE)** - Mean of the squared value of errors (squared distance from true value): In statistics, the mean squared error or mean squared deviation (MSD) of an estimator which measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. MSE is a risk function, corresponding to the expected value of the squared error loss. The fact that MSE is almost always strictly positive (and not zero) is because of randomness or because the estimator does not account for information that could produce a more accurate estimate.

$$\text{MSE} = \underbrace{\frac{1}{n}\sum_{i=1}^{n}}_{\text{test set}} \underbrace{(y_i}_{\text{predicted vaue}} - \underbrace{\hat{y}_i}_{\text{actual value}})^2$$

- **R^2 (coefficient of determination)** - Regression score function: Coefficient of determination also called as R2 score is used to evaluate the performance of a linear regression model. It is the amount of the variation in the output dependent attribute which is predictable from the input independent variable(s). It is used to check how well-observed results are reproduced by the model, depending on the ratio of total deviation of results described by the model.

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}$$

5.4.1 Result Analysis

	models	MAE	MSE	R^2	Cross Validation Score
2	Lasso Regressor	838.07	1285554.86	0.5594	0.5581
0	Linear Regression	838.20	1285809.57	0.5593	0.5580
1	Random Forest Regressor	1030.27	1964025.66	0.3268	0.5920

Here, we compare different models of regressions, i.e., Lasso regressor, Linear regressor and the Random Forest Regressor. From the above table, we see that Mean Absolute Error(MAE) and the Mean Squared Error(MSE) is the Most in the Random Forest Regressor when compared to the Linear Regressor and the Lasso Regressor. Here the R^2 (coefficient of determination) which is the regression score function is lower of the random forest regressor when compared to the Lasso regressor and the Linear Regressor.

Linear Regression and Lasso Regressor have the best perfomance in most categories. The performance of the Random Forest is not optimal even though his cross validation is the highest. Here, from the result, we can see that the Linear Regressor and the Lasso Regressor can be to used in our Big MART sales application for the prediction of the sales. Further we noted that by using the Linear Regressor and the Lasso Regressor, the accuracy of the prediction of sales also increased when compared to the Random Forest Regressor.

By using this data prediction models, we are successful in predicting the sales of the products in the particular stores which helps both the Stores manager and the Customer. Here, from the result, we can see that the Linear Regressor and the Lasso Regressor can be to used in our Big MART sales application for the prediction of the sales. Further we noted that by using the Linear Regressor and the Lasso Regressor, the accuracy of the prediction of sales also increased when compared to the Random Forest Regressor.

5.4.2 Output Screens

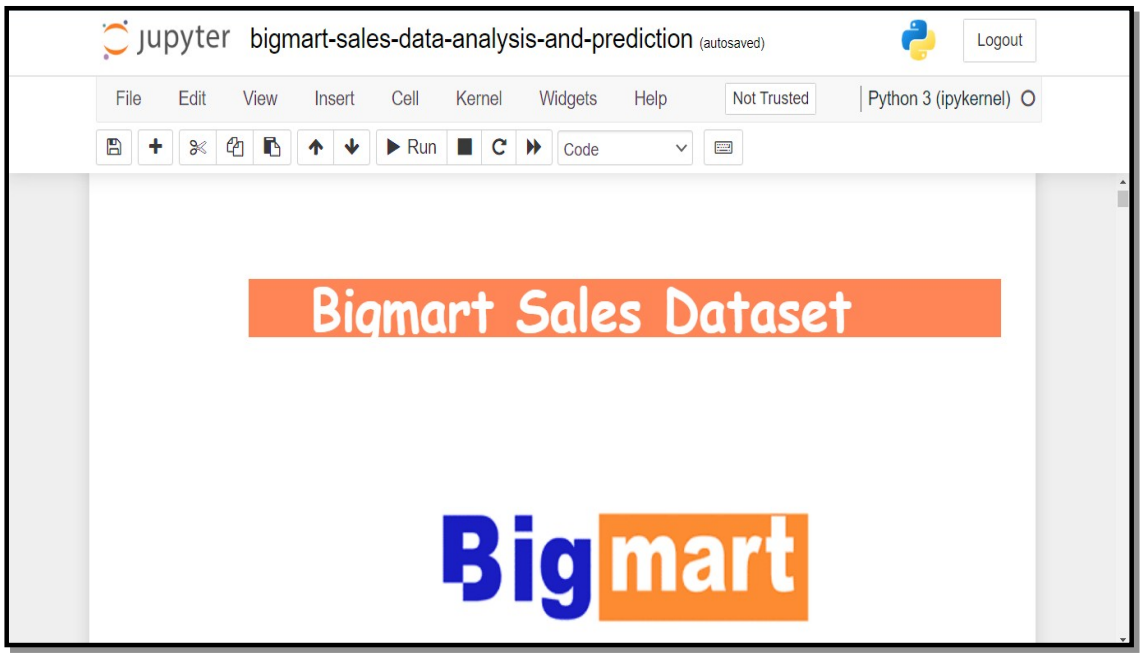
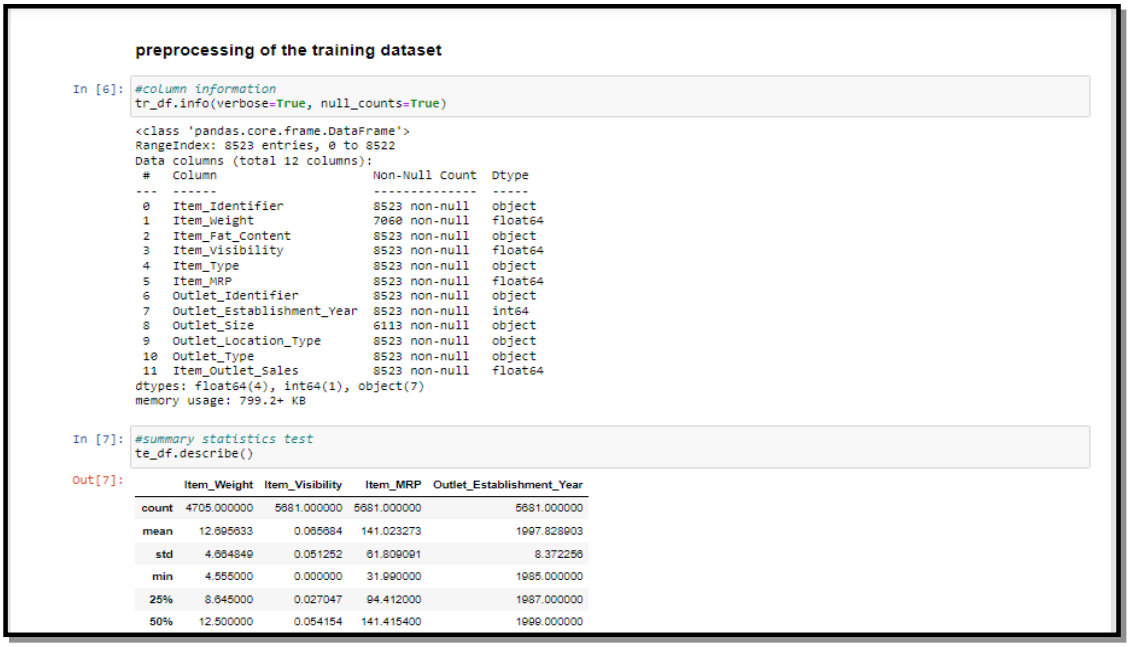


Fig 5.4.2: Here this the homescreen on the bigmart sales prediction project.



preprocessing of the training dataset

In [6]:
```
#column information
tr_df.info(verbose=True, null_counts=True)
```

```
<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
```

In [7]:
```
#summary statistics test
te_df.describe()
```

Out[7]:

	Item_Weight	Item_Visibility	Item_MRP	Outlet_Establishment_Year
count	4705.000000	5681.000000	5681.000000	5681.000000
mean	12.695633	0.065684	141.023273	1997.828903
std	4.664849	0.051252	61.809091	8.372256
min	4.555000	0.000000	31.990000	1985.000000
25%	8.645000	0.027047	94.412000	1987.000000
50%	12.500000	0.054154	141.415400	1999.000000

Fig 5.4.2.1: here in the above figure we can see the preprocessing of the dataset

Data Visualization 📊

Univariate Plots

For starters we will create countplots for the categorical columns:

In [20]:
```
#categorical columns:
['Item_Identifier', 'Item_Fat_Content', 'Item_Type', 'Outlet_Identifier',
 'Outlet_Size', 'Outlet_Location_Type', 'Outlet_Type']

plt.figure(figsize=(6,4))
sns.countplot(x='Item_Fat_Content' , data=tr_df ,palette='mako')
plt.xlabel('Item_Fat_Content', fontsize=14)
plt.show()
```

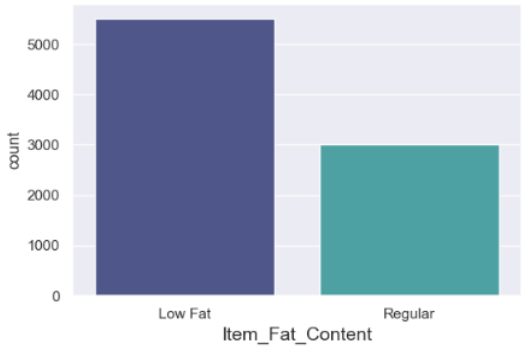


Fig 5.4.2.2: Data Visualization with Univariate Plots

multivariate plots

I want to check the following relationships with `Item_Outlet_Sales` :

- Sales per item type
- Sales per outlet
- Sales per outlet type
- Sales per outlet size
- Sales per location type

In [28]:
```
plt.figure(figsize=(27,10))
sns.barplot('Item_Type' ,'Item_Outlet_Sales', data=tr_df ,palette='gist_rainbow_r')
plt.xlabel('Item_Type', fontsize=14)
plt.legend()
plt.show()
```

```
No artists with labels found to put in legend.  Note that artists whose label start with an underscore are ignored when legend
() is called with no argument.
```

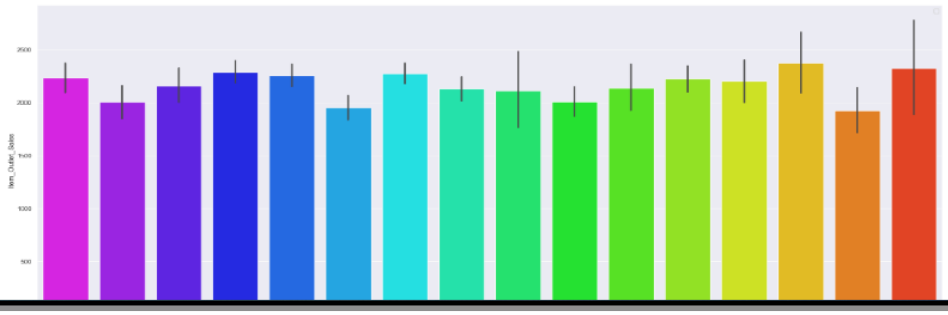
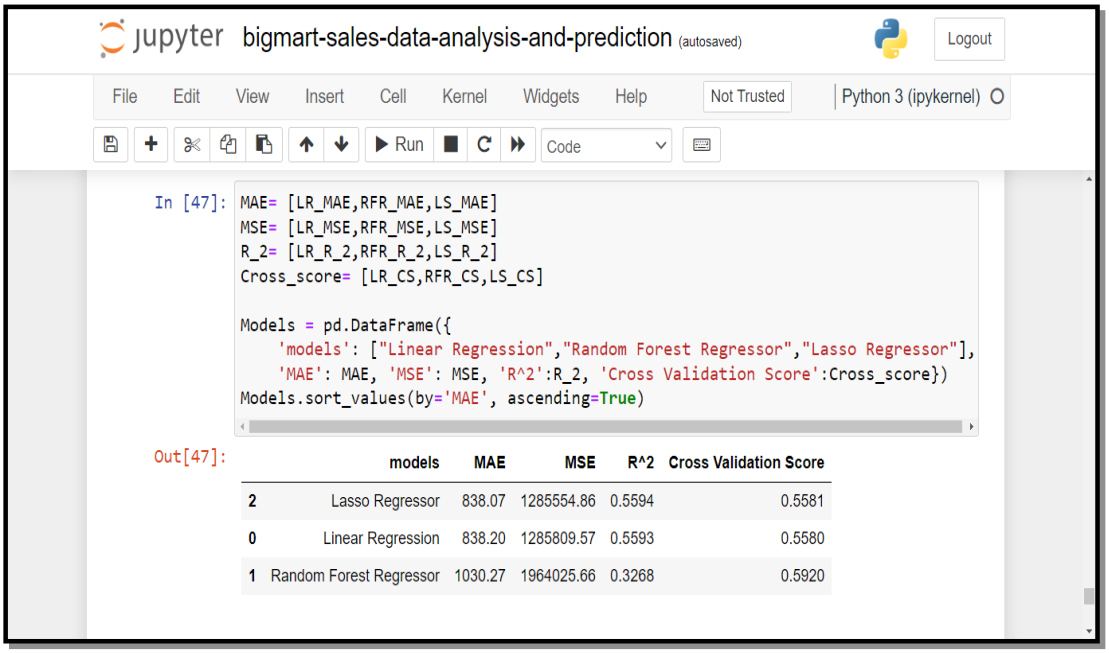


Fig 5.4.2.3: Data Visualization with multivariate plots



	models	MAE	MSE	R^2	Cross Validation Score
2	Lasso Regressor	838.07	1285554.86	0.5594	0.5581
0	Linear Regression	838.20	1285809.57	0.5593	0.5580
1	Random Forest Regressor	1030.27	1964025.66	0.3268	0.5920

Fig 5.4.2.4: Comparing of results from different predicting models

TESTING

6.1 Introduction

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 Testing Methodologies

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration test

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Valid Input : identified classes of valid input must be accepted.
- Invalid Input : identified classes of invalid input must be rejected.
- Functions : identified functions must be exercised.
- Output : identified classes of application outputs must be exercised.
- Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. You cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software life cycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

6.3 Design of test cases and scenarios

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

6.4 VALIDATION

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures

caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.
Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.
Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CONCLUSION

In this paper, basics of machine learning and the associated data process and modeling algorithms have been described, followed by their application for the task of sales prediction in Big Mart shopping centers at different locations. On implementation, the prediction results show the correlation among different attributes considered and how a particular location of medium size recorded the highest sales, suggesting that other shopping locations should follow similar patterns for improved sales. Multiple instances parameters and various factors can be used to make this sales prediction more innovative and successful. Accuracy, which plays a key role in prediction-based systems, can be significantly increased as the number of parameters used are increased. Also, a look into how the sub-models work can lead to increase in productivity of system. One of the main highlights is more expressive regression outputs, which are more understandable bounded with some of accuracy. Moreover, the flexibility of the proposed approach can be increased with variants at a very appropriate stage of regression model-building

7.1 Future Enhancements

The project can be further collaborated in a web - based application or in any device supported with an in-built intelligence by virtue of Internet of Things (IoT), to be more feasible for use. Various stakeholders concerned with sales information can also provide more inputs to help in hypothesis generation and more instances can be taken into consideration such that more precise results that are closer to real world situations are generated.

Moreover, the flexibility of the proposed approach can be increased with variants at a very appropriate stage of regression model-building. There is a further need of experiments for proper measurements of both accuracy and resource efficiency to assess and optimize correctly.

BIBLIOGRAPHY

[1] Smola, A., & Vishwanathan, S. V. N. (2008). Introduction to machine learning. Cambridge University, UK, 32, 34.

[2] Saltz, J. S., & Stanton, J. M. (2017). An introduction to data science. Sage Publications.

[3] Shashua, A. (2009). Introduction to machine learning: Class notes 67577. arXiv preprint arXiv:0904.3664.

[4] MacKay, D. J., & Mac Kay, D. J. (2003). Information theory, inference and learning algorithms. Cambridge university press.

[5] Daumé III, H. (2012). A course in machine learning. Publisher, ciml. Info, 5, 69.

[6] Cerrada, M., & Aguilar, J. (2008). Reinforcement learning in system identification. In Reinforcement Learning. IntechOpen.

[7] Welling, M. (2011). A first encounter with Machine Learning. Irvine, CA.: University of California, 12.

[8] Learning, M. (1994). Neural and Statistical Classification. Editors D. Mitchie et. al, 350.

[9] Mitchell, T. M. (1999). Machine learning and data mining. Communications of the ACM, 42(11), 30-36.

[10] Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media