

TARGET CORPORATION SALES PREDICTION

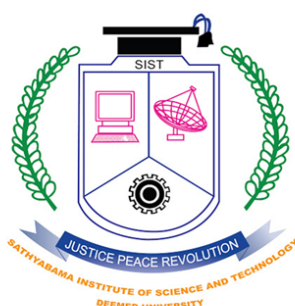
Submitted in partial fulfillment of the requirements for the award of

Bachelor of Engineering Degree in
Computer Science and Engineering

By

Snigdha Saha (38110546)

Shivani Agarwal (38110526)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY) Accredited with Grade "A" by NAAC I 12B

Status by UGC I Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

March – 2022



SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)



(Established under Section 3 of UGC Act, 1956)
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600119
www.sathyabamauniversity.ac.in

SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Snigdha Saha (38110546)**, **Shivani Agarwal (38110526)** carried out the project entitled “**Target Corporation Sales Prediction**” under our supervision from ___to___.

Internal Guide

Dr. Veena K.

Head of the Department

Dr. S.VIGNESHWARI, M.E., Ph.D.,

Dr. LAKSHMANAN L, M.E., Ph.D.

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

We, **SHIVANI AGARWAL** (38110526) and **SNIGDHA SAHA** (38110546) hereby declare that the Project Report entitled “**TARGET CORPORATION SALES PREDICTION**” done by me under the guidance of Dr.Veena K. is submitted at SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the Board of **Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D., Dean**, School of Computing , **Dr.S.Vigneshwari M.E., Ph.D., and Dr.L.Lakshmanan M.E., Ph.D.**, Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.Veena K.** for his valuable guidance, suggestions and constant encouragement paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **DEPARTMENT OF COMPUTING** who were helpful in many ways for the completion of my project.

ABSTRACT

Nowadays, competition among business corporations is very high and supermarkets are one such type of business corporation. The most important factor for supermarkets to stay in the competition is their customers' satisfaction. Providing the customers with the variety of products they need and also the required quantity under one roof is a big task and to achieve that corporations like Target Corporation, which has several stores across the globe keep track of every product's sales data. This data store contains the attributes of various items and also the individual customer's data which is then used to predict potential consumer demand and fulfill their consumer's needs as required. Anomalies and general trends are often discovered by mining the data warehouse's data store. For retailers like Target, the resulting data can be used to predict future sales volume using various machine learning techniques. Therefore, a predictive model was developed using Xgboost, Linear regression, and Ridge regression techniques for forecasting the sales of a business such as Target Corporation, and it was discovered that the model outperforms existing models. It was able to give a brief view of the number of items sold per product.

TABLE OF CONTENTS

Chapter No.	Title	Page No.
	Abstract	6
	List of Figures	9
1.	INTRODUCTION Outline Of The Project Purpose Of The Project Problem In Existing System Proposed System	10
2.	AIM & SCOPE OF THE PRESENT INVESTIGATION Workflow Diagram System Architecture Block Diagram	13
3.	PROJECT IMPLENTATIONS, ALGORITHMS AND METHODS USED Introduction Hardware Requirement Software Requirement Working Explanation	15
4.	IMPLEMENTED SCREENSHOTS	29

5.	SUMMARY & CONCLUSION	32
6.	APPENDIX A. SOURCE CODE REFERENCES	33 37

LIST OF FIGURES

Figure No.	Title	Page No.
2.1	Workflow Diagram	14
2.2	System Architecture Diagram	15
3.1	Categorical Features Graph	23,24
3.2.1	Numerical Features Graph	25
3.2.2	Outlet Graphs	26
3.2.3	Outlet graphs	27

CHAPTER I

INTRODUCTION:

Outline of the project:

Everyday competitiveness between various shopping centers as and as huge marts is becoming higher intense, violent just because of the quick development of global malls also online shopping. Each market seeks to offer personalized and limited-time deals to attract many clients relying on period of time, so that each item's volume of sales may be estimated for the organization's stock control, transportation and logistical services. The current machine learning algorithm is very advanced and provides methods for predicting or forecasting sales any kind of organization, extremely beneficial to overcome low – priced used for prediction. Always better prediction is helpful, both in developing and improving marketing strategies for the marketplace, which is also particularly helpful. In today's modern world, huge shopping centers such as big malls and marts are recording data related to sales of items or products with their various dependent or independent factors as an important step to be helpful in prediction of future demands and inventory management. The dataset built with various dependent and independent variables is a composite form of item attributes, data gathered by means of customer, and also data related to inventory management in a data warehouse. The data is thereafter refined in order to get accurate predictions and gather new as well as interesting results that shed a new light on our knowledge with respect to the task's data. This can then further be used for forecasting future sales by means of employing machine learning algorithms such as the random forests and simple or multiple linear regression model.

● Purpose of the Project:

An accurate sales forecast allows you to properly plan for impending sales. If you have developed an accurate understanding of how your business' sales naturally increase and decrease over the year, you can plan by:

- Keeping appropriate levels of stock.
- Hiring temporary staff

- Spacing out projects that would otherwise preoccupy staff needed to support an increase in demand

● **Proposed System:**

In this paper, we are addressing the problem of Target mart sales prediction or forecasting of an item on customer's future demand in different target mart stores across various locations and products based on the previous record. We started with making some hypothesis about the data without looking at it. Then we moved on to data exploration where we found out some nuances in the data which required remediation. Next, we performed data cleaning and feature engineering, where we imputed missing values and solved other irregularities, made new features and also made the data model-friendly. Finally we made regression, decision tree and random forest model and got a glimpse of how to tune them for better results.

● **Advantages:**

1. Sales forecasting helps you attain this revenue efficiency by offering insight into the likely behavior of your most valuable customers. You can predict future sales, as well as improve pricing, advertising, and product development.
2. Forecasts help revenue teams achieve their goals by identifying early warning signals in their pipeline and course-correcting before it's too late.
3. Quantifies your organization's health.
Your forecast is much more than a sales number, just like revenue operations includes marketing and customer success in addition to sales.
4. Enables continuous strategic planning.
According to recent Forrester research, annual or even quarterly planning is a thing of the past. Continuous forecasting enables organizations to respond with exponentially less failure to today's pace

of the market.

- **Disadvantages:**

1. Time-Intensive Completion - While there are various methods of sales forecasting, the two broad approaches include manual and data-driven processes. In either case, significant time is required to develop forecasts.
2. Internal Bias - Forecasting is not always intended to be a realistic projection of anticipated sales and not a depiction of desired sales. The challenge for company marketing and sales reps in preparing forecasts is that internal bias is hard to avoid.
3. Sales reps look better and tend to earn more commission when they achieve high sales goals. This natural desire to have lofty aspirations can lead to inflated forecasts. When sales forecasts are high, companies could invest too much in inventory and resources in preparation for selling activities.

CHAPTER II

LITERATURE SURVEY:

A great deal of work having been gotten really intended to date the territory of deals foreseeing. A concise audit of the important work in the field of big mart deals is depicted in this part. Numerous other Measurable methodologies, for example, with regression, (ARIMA) Auto-Regressive Integrated Moving Average, (ARMA) Auto-Regressive Moving Average, have been utilized to develop a few deals forecast standards. Be that as it may, deals anticipating is a refined issue and is influenced by both outer and inside factors, and there are two significant detriments to the measurable technique as set out in A. S. Weigend et A mixture occasional quantum relapse approach and (ARIMA) Auto-Regressive Integrated Moving Average way to deal with every day food deals anticipating were recommend by N. S. Arunraj and furthermore found that the exhibition of the individual model was moderately lower than that of the crossover model.

E. Hadavandi utilized the incorporation of “Genetic Fuzzy Systems (GFS)” and information gathering to conjecture the deals of the printed circuit board. In their paper, K-means bunching delivered K groups of all information records. At that point, all bunches were taken care of into autonomous with a data set tuning and rule-based extraction ability. Perceived work in the field of deals gauging was done by P.A. Castillo, Sales estimating of new distributed books was done in a publication market the executives setting utilizing computational techniques. “Artificial neural organizations” are additionally utilized nearby income estimating. Fluffy Neural Networks have been created with the objective of improving prescient effectiveness, and the Radial “Base Function Neural Network (RBFN)” is required to have an incredible potential for anticipating deals.

A Forecast for Big Mart Sales Based on Random Forests and Multiple Linear Regression (2018) Author: - Kadam, H., Shevade, R., Ketkar, P. and Rajguru Description: - A Forecast for Big Mart Sales Based on Random Forests and Multiple Linear Regression used Random Forest and Linear Regression for prediction

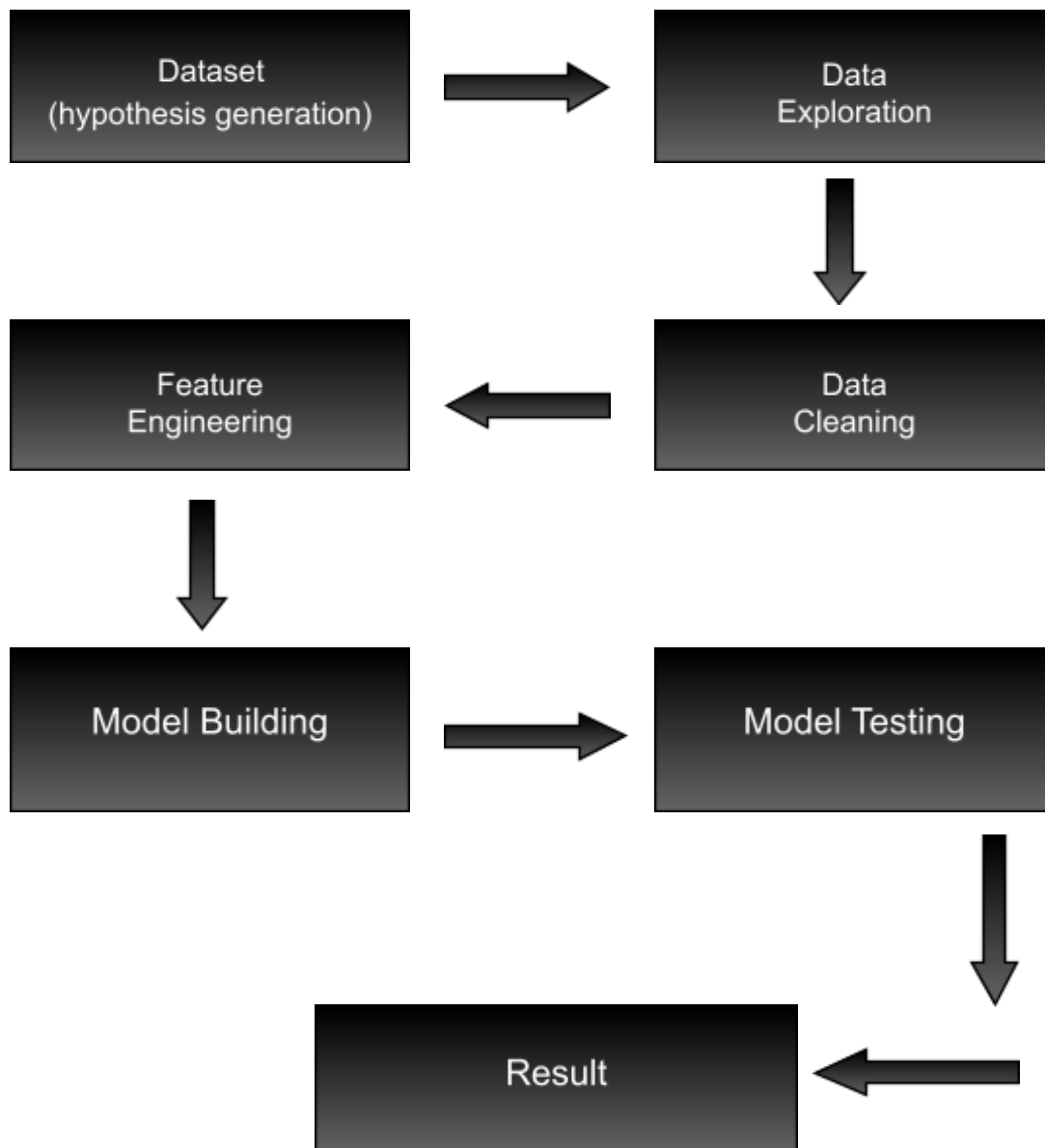
analysis which gives less accuracy. To overcome this we can use XG boost Algorithm which will give more accuracy and will be more efficient.

Forecasting methods and applications (2008) Author: - Makridakis, S., Wheelwrigth.S.C., Hyndman. R.J Description: - Forecasting methods and applications contains Lack of Data and short life cycles. So some of the data like historical data, consumer-oriented markets face uncertain demands, can be prediction for accurate result. 3. Title: -Comparison of Different Machine Learning Algorithms for Multiple Regression on Black Friday Sales Data (2018) Author: - C. M. Wu, P. Patil and S. Gunaseelan Description: - Comparison of Different Machine Learning Algorithms for Multiple Regression on Black Friday Sales Data Used Neural Network for comparison of different algorithms. To overcome this Complex models like neural networks is used for comparison between different algorithms which is not efficient so we can use more simpler algorithm for prediction

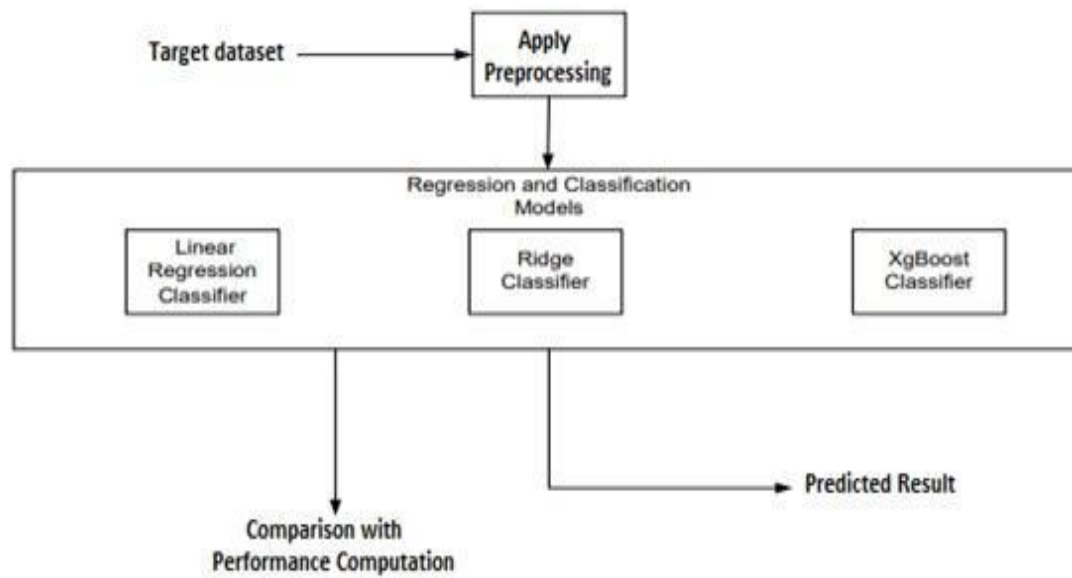
CHAPTER THREE

PROJECT DESIGN:

- 1.Workflow Diagram:



- **2.System Architecture:**



CHAPTER III

PROJECT IMPLEMENTATION, ALGORITHMS AND METHODS USED:

● Introduction:

“To find out what role certain properties of an item play and how they affect their sales by understanding Target mart sales.” In order to help Target Mart achieve this goal, a predictive model can be built to find out for every store, the key factors that can increase their sales and what changes could be made to the product or store’s characteristics.

● Software Requirements:

Windows XP (8,9 or 10).

Any Python Interpreter.

Machine Learning modules installed.

Target Mart Dataset.

● Working Explanation:

The aim is to build a predictive model and find out the sales of each product at a particular store. Using this model, Big Marts like Target will try to understand the properties of products and stores which play a key role in increasing sales. So the idea is to find out the properties of a product, and store which impacts the sales of a product. We came up with certain hypothesis in order to solve the problem statement relating to the various factors of the products and stores. We develop a predictive model using different ML Algorithms like Linear regression, Polynomial regression, and Ridge regression techniques for forecasting the sales of a business such as Target Corporation.

- By using this model, we will try to understand the properties of products and stores which play a key role in increasing sales.
- We came up with certain hypothesis in order to solve the problem statement relating to the various factors of the products and stores.

- We'll be performing some basic data exploration and come up with some inferences about the data.
- In our model we have used Target corporation sales dataset. After preprocessing and filling missing values, we used ensemble classifier using Decision trees, Linear regression, Ridge regression, Polynomial regression and XgBoost regression classifier.

• Prerequisites:

The dataset contains about 10 years of daily weather observations from numerous Australian weather stations.

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by meta programming and meta objects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Python's design offers some support for functional programming in the Lisp tradition. It has functions; list comprehensions, dictionaries, sets, and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

Regression is a statistical measure that attempts to determine the strength of the relationship between one dependent variable usually denoted by Y and a series of other changing variables known as independent variables. Regression model which contain more than two predictor variables are called Multiple Regression Model.

• Model Design:

This shows the architecture Diagram of the proposed model where they focus on the different algorithm application to the dataset. Here, we are calculating the Accuracy, MAE, MSE, RMSE and final concluding the best yield algorithm. Here are the following Algorithm are used.

A. Linear Regression:

- Build a fragmented plot:

1) a linear or non-linear pattern of data and 2) a variance (outliers).

Consider a transformation if the marking isn't linear. If this is the case, outsiders, it can suggest only eliminating them if there is a non-statistical justification.

- Link the data to the least squares line and confirm the model assumptions using the residual plot (for the constant standard deviation assumption) and the normal probability plot (for the normal probability assumption) A transformation might be necessary if the assumptions made do not appear to be met

- If required, convert the data to the least square using the transformed data, construct a regression line.

- If a change has been completed, return to the previous process 1. If not, continue to phase 5.

- When a "good-fit" classic is defined, write the least-square regression line equation. Consist of normal estimation, estimation, and R-squared errors. • Linear regression formulas look like this:

$$Y = o_1x_1 + o_2x_2 + \dots + o_nx_n$$

R-Square: Defines the difference in X (depending variable) explains the total variance in Y (dependent variable) (independent variable).

B. Polynomial Regression:

- Polynomial Regression is a relapse calculation that modules the relationship here among dependent(y) and the autonomous variable(x) in light of the fact that as most extreme limit polynomial. The condition for polynomial relapse is given beneath: $y = b_0 + b_1x_1 + b_2x_1^2 + b_3x_1^3 + \dots + b_nx_1^n$

- It is regularly alluded to as the exceptional instance of various straight relapse in ML. Since we apply some polynomial terms to the numerous

straight relapse condition to change it to polynomial relapse adjustment to improve accuracy.

- The informational collection utilized for preparing in polynomial relapse is of a non-straight nature.
- It uses a linear regression model to fit complex and non-linear functions and datasets.

C. Ridge Regression:

Ridge regression is a model tuning tool used to evaluate any data that suffers from multi-collinearity. This method performs the L2 regularization procedure. When multi-collinearity issues arise, the least squares are unbiased and the variances are high, resulting in the expected values being far removed from the actual values. The cost function for ridge regression:

$$\text{Min} (||Y - X(\theta)||^2 + \lambda ||\theta||^2)$$

D. XgBoost Regression:

“Extreme Gradient Boosting” is same but much more effective to the gradient boosting system. It has both a linear model solver and a tree algorithm. Which permits “XgBoost” in any event multiple times quicker than current slope boosting executions? It underpins various target capacities, including relapse, order and rating. As "XgBoost" is extremely high in prescient force however generally delayed with organization, it is appropriate for some rivalries. It likewise has extra usefulness for cross-approval and finding significant factors.

● Methodology:

We will explore the problem in following stages:

1. **Hypothesis Generation** - understanding the problem better by brainstorming possible factors that can impact the outcome.
2. **Data Preprocessing** - looking at categorical and continuous feature summaries and making inferences about the data.
3. **Data Analysis** - imputing missing values in the data and checking for outliers.

4. **Feature Engineering** - modifying existing variables and creating new ones for analysis.
5. **Implementing model** - making predictive models on the data.

1. Importing Libraries:

The first step in any Data Analysis step is importing necessary libraries.

Load Data Set: Dataset can be loaded using a method `read_csv()`.

2. Data Preprocessing:

Real-world data is

often messy, incomplete, unstructured, inconsistent, redundant, sprinkled with wacky values. So, without deploying any Data Preprocessing techniques, it is almost impossible to gain insights from raw data. Data preprocessing is a process of converting raw data to a suitable format to extract insights. It is the first and foremost step in the Data Science life cycle. Data Preprocessing makes sure that data is clean, organize and read-to-feed to the Machine Learning model.

- Dataset has two data types: float64, object
- Except for the Date, Location columns, every column has missing values.

Let's generate descriptive statistics for the dataset using the function `describe()` in pandas.

Descriptive Statistics: It is used to summarize and describe the features of data in a meaningful way to **extract insights**. It uses two types of statistic to describe or summarize data:

- Measures of tendency
- Measures of spread.

3. Cardinality check for Categorical features:

The accuracy, performance of a classifier not only depends on the model that we use, but also depends on how we preprocess data, and what kind of data you're feeding to the classifier to learn.

Many Machine learning algorithms like Linear Regression, Logistic Regression, k-nearest neighbors, etc. can handle only numerical data, so encoding categorical data to numeric becomes a necessary step. But before jumping into encoding, check the cardinality of each categorical feature.

Cardinality: The number of unique values in each categorical feature is known as cardinality.

A feature with a high number of distinct/ unique values is a high cardinality feature. A categorical feature with hundreds of zip codes is the best example of a high cardinality feature.

This high cardinality feature poses many serious problems like it will increase the number of dimensions of data when that feature is encoded. This is not good for the model.

There are many ways to handle high cardinality, one would be **feature engineering** and the other is simply dropping that feature if it doesn't add any value to the model.

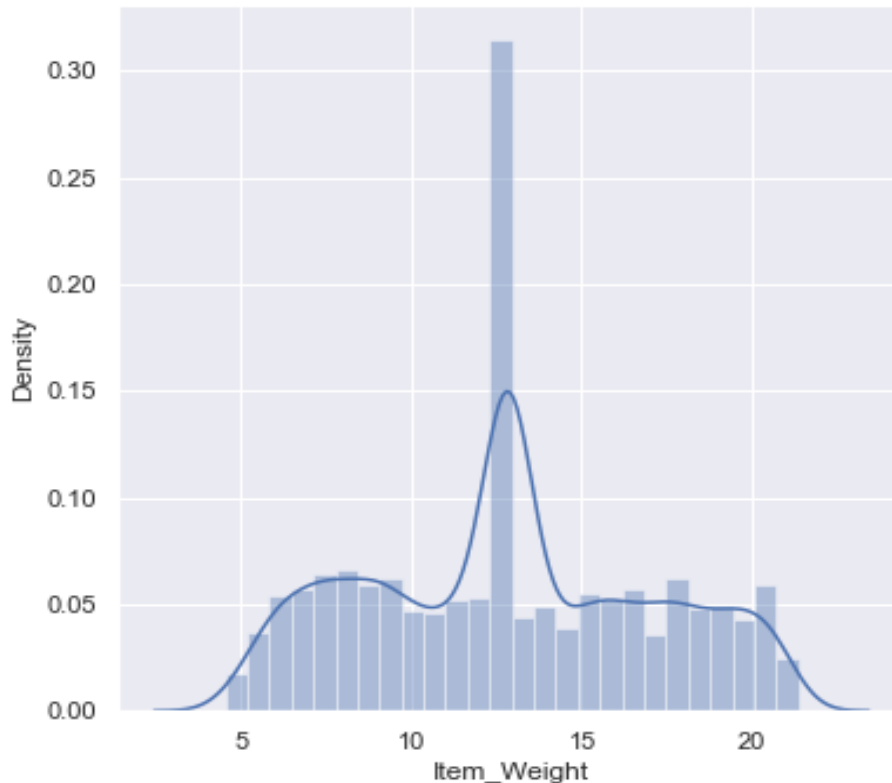
4. Handling Missing Values:

Machine learning algorithms can't handle missing values and cause problems. So they need to be addressed in the first place. There are many techniques to identify and impute missing values.

If a dataset contains missing values and loaded using pandas, then missing values get replaced with NaN(Not a Number) values. These NaN values can be identified using methods like **isna()** or **isnull()** and they can be imputed using **fillna()**. This process is known as **Missing Data Imputation**.

Handling Missing values in Categorical Features:

```
big_mart_data.isnull().sum()
```



Missing values in Numerical Features can be imputed using Mean and Median. Mean is sensitive to outliers and median is immune to outliers. If you want to impute the missing values with mean values, then outliers in numerical features need to be addressed properly.

5. Outliers detection and treatment:

An Outlier is an observation that lies an abnormal distance from other values in a given sample. They can be detected using visualization (like box-plots, scatter plots), Z-score, statistical and probabilistic algorithms, etc.

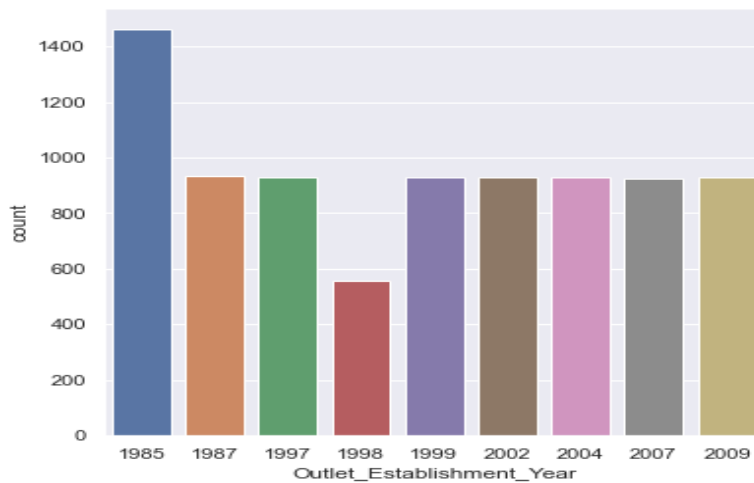
It's time to do some analysis on each feature to understand about data and get some insights.

6. Exploratory Data Analysis:

Exploratory Data Analysis(EDA) is a technique used to analyze, visualize, investigate, interpret, discover and summarize data. It helps Data Scientists to extract trends, patterns, and relationships in data.

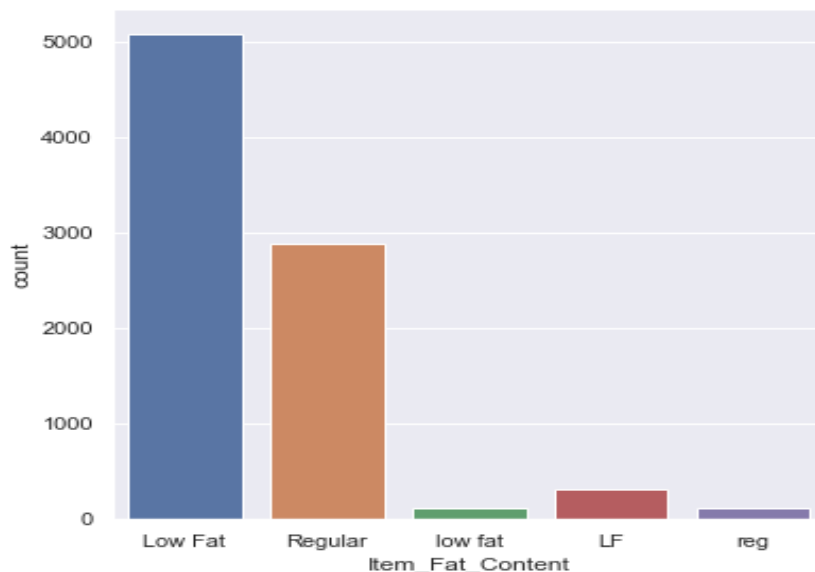
Outlet_Establishment_Year column:

```
plt.figure(figsize=(6,6))
sns.countplot(x='Outlet_Establishment_Year', data=big_mart_data)
plt.show()
```



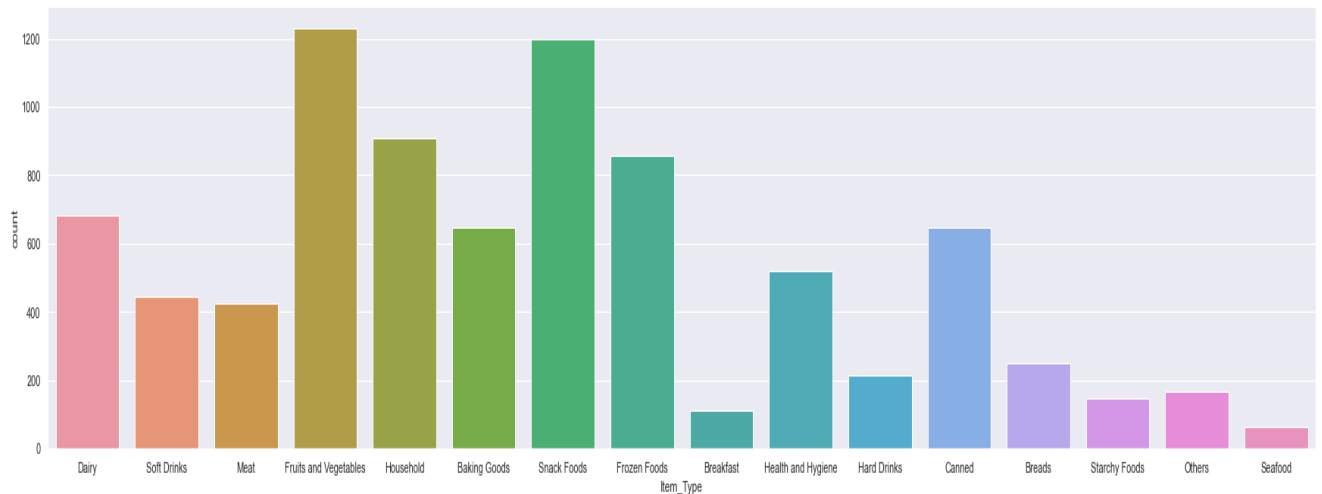
Item_Fat_Content column:

```
plt.figure(figsize=(6,6))
sns.countplot(x='Item_Fat_Content', data=big_mart_data)
plt.show()
```



Item_Type column:

```
plt.figure(figsize=(30,6))
sns.countplot(x='Item_Type', data=big_mart_data)
plt.show()
```



7. Encoding of Categorical Features:

Most Machine Learning Algorithms like Logistic Regression, Support Vector Machines, K Nearest Neighbours, etc. can't handle categorical data. Hence, these categorical data need to be converted to numerical data for modeling, which is called Feature Encoding.

There are many feature encoding techniques like One code encoding, label encoding. But in this particular blog, I will be using `replace()` function to encode categorical data to numerical data.

8. Correlation:

Correlation is a statistic that helps to measure the strength of the relationship between two features. It is used in bivariate analysis. Correlation can be calculated with method ***corr()*** in pandas.

Splitting data into Independent Features and Dependent Features:

For feature importance and feature scaling, we need to split data into independent and dependent features.

- *X* – Independent Features or Input features

- y – Dependent Features *or* target label.

9. Feature Importance:

- Machine Learning Model performance depends on features that are used to train a model. **Feature importance** describes which features are relevant to build a model.
- **Feature Importance** refers to the techniques that assign a score to input/label features based on how useful they are at predicting a target variable. Feature importance helps in **Feature Selection**.

We'll be using **ExtraTreesRegressor** class for Feature Importance. This class implements a meta estimator that fits a number of randomized decision trees on various samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

10. Feature Scaling:

Feature Scaling is a technique used to scale, normalize, standardize data in range(0,1). When each column of a dataset has distinct values, then it helps to scale data of each column to a common level. **StandardScaler** is a class used to implement feature scaling.

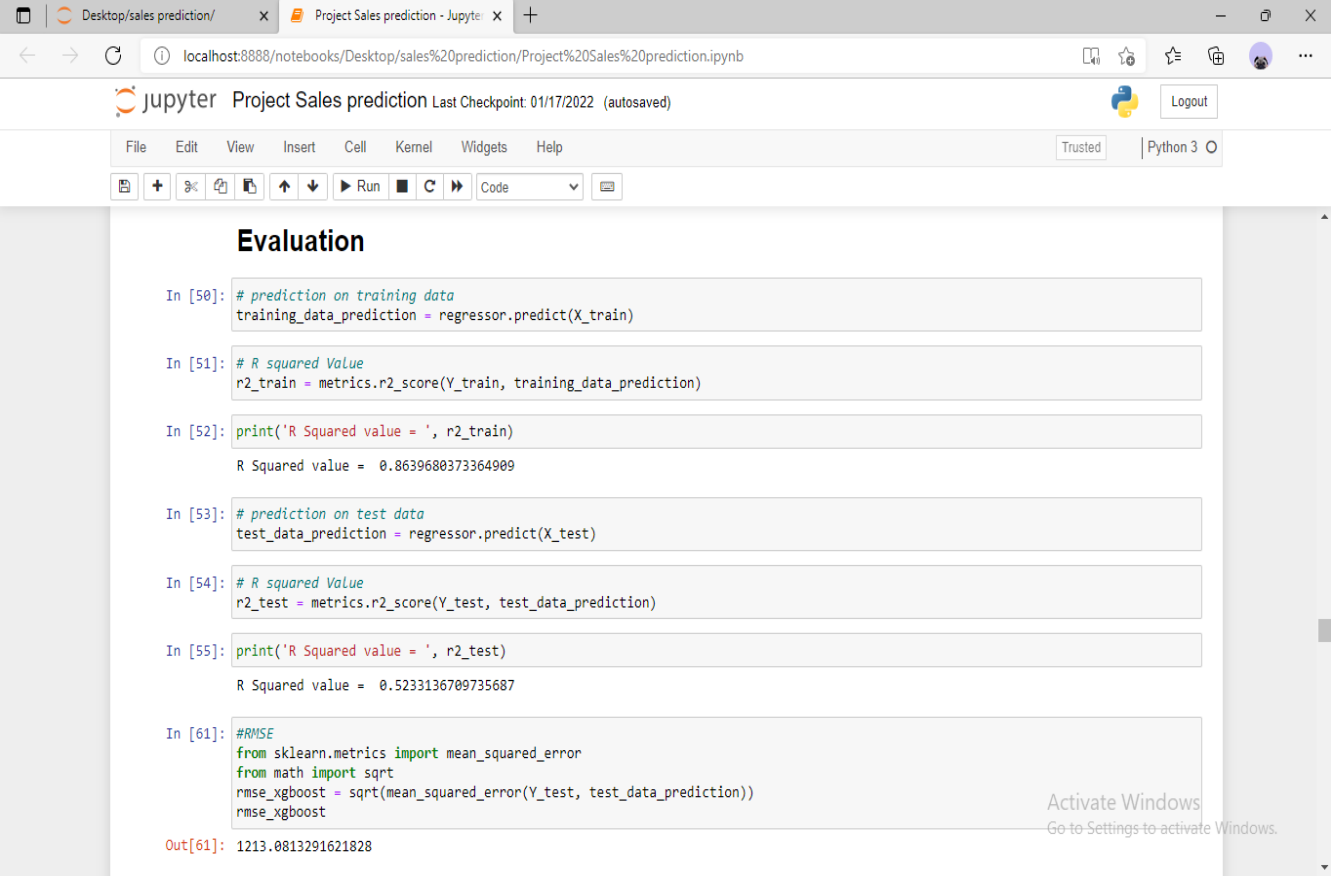
11. Model Building:

In this article, I will be using a **Logistic Regression** algorithm to build a predictive model to predict whether or not it will rain tomorrow.

CHAPTER IV

IMPLEMENTED SCREENSHOTS:

1.



The screenshot shows a Jupyter Notebook interface in a web browser. The browser tabs include 'Desktop/sales prediction/' and 'Project Sales prediction - Jupyter:'. The address bar shows 'localhost:8888/notebooks/Desktop/sales%20prediction/Project%20Sales%20prediction.ipynb'. The Jupyter Notebook header displays 'Project Sales prediction' and 'Last Checkpoint: 01/17/2022 (autosaved)'. The notebook has a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. Below the menu bar is a toolbar with icons for file operations, running, and code execution. The notebook content is titled 'Evaluation' and contains several code cells. The first cell (In [50]) predicts training data. The second cell (In [51]) calculates the R-squared value for training data, which is printed in the third cell (In [52]) as 0.8639680373364909. The fourth cell (In [53]) predicts test data. The fifth cell (In [54]) calculates the R-squared value for test data, which is printed in the sixth cell (In [55]) as 0.5233136709735687. The seventh cell (In [61]) calculates the RMSE, with the output (Out[61]) being 1213.0813291621828. An 'Activate Windows' watermark is visible in the bottom right corner of the notebook area.

```
Evaluation

In [50]: # prediction on training data
training_data_prediction = regressor.predict(X_train)

In [51]: # R squared Value
r2_train = metrics.r2_score(Y_train, training_data_prediction)

In [52]: print('R Squared value = ', r2_train)
R Squared value = 0.8639680373364909

In [53]: # prediction on test data
test_data_prediction = regressor.predict(X_test)

In [54]: # R squared Value
r2_test = metrics.r2_score(Y_test, test_data_prediction)

In [55]: print('R Squared value = ', r2_test)
R Squared value = 0.5233136709735687

In [61]: #RMSE
from sklearn.metrics import mean_squared_error
from math import sqrt
rmse_xgboost = sqrt(mean_squared_error(Y_test, test_data_prediction))
rmse_xgboost

Out[61]: 1213.0813291621828
```

2.

The screenshot shows a Jupyter Notebook titled "Project Sales prediction" with the following code and outputs:

```
In [68]: #Independent Variables:
x = new_data.drop("item_outlet_sales", axis = 1)

#Depenedent Variables
y = new_data["item_outlet_sales"].values.reshape(-1,1)

In [69]: #Splitting The data into Train and Test Dataset:
from sklearn.model_selection import train_test_split
x_train,x_test, y_train, y_test = train_test_split(x,y, test_size =0.20, random_state = 3)

In [70]: #Applying Linear Regression Model
from sklearn.linear_model import LinearRegression
regressor =LinearRegression()
regressor.fit(x_train, y_train)

Out[70]: LinearRegression()

In [71]: #Prediction
y_pred = regressor.predict(x_test)

In [72]: #Accuracy of Model (Apply R2_score)
from sklearn.metrics import r2_score, mean_squared_error
r2_score(y_test, y_pred)

Out[72]: 0.5370776070748289

In [74]: #Checking Root Mean Square error
from math import sqrt
rmse = sqrt(mean_squared_error(y_test, y_pred))
rmse

Out[74]: 1174.0872537355372
```

Activate Windows
Go to Settings to activate Windows.

3.

The screenshot shows a Jupyter Notebook titled "Project Sales prediction" with the following code and outputs:

```
Ridge Regressor

In [76]: #Ridge Regression
from sklearn.linear_model import Ridge
rr = Ridge(alpha = 0.009)
rr.fit(x_train, y_train)

Out[76]: Ridge(alpha=0.009)

In [77]: rr_pred = rr.predict(x_test)

In [78]: #Accuracy score check
r2_score(y_test, y_pred)

Out[78]: 0.5370776070748289

In [79]: #RMSE
rmse_ridge = sqrt(mean_squared_error(y_test, rr_pred))
rmse_ridge

Out[79]: 1174.0870771007833

In [ ]:
```

Activate Windows
Go to Settings to activate Windows.

CHAPTER V

CONCLUSION:

In this work, the effectiveness of various algorithms on the data on revenue and review of, best performance-algorithm, here propose a software to using regression approach for predicting the sales centered on sales data from the past the accuracy of linear regression prediction can be enhanced with this method, polynomial regression, Ridge regression, and Xgboost regression can be determined. So, we can conclude ridge and Xgboost regression gives the better prediction than the Linear and polynomial regression approaches. In future, the forecasting sales and building a sales plan can help to avoid unforeseen cash flow and manage production, staff and financing needs more effectively. In future work we can also consider with the ARIMA model which shows the time series graph.

APPENDIX

SOURCE CODE:

#Importing Libraries:

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.preprocessing import LabelEncoder

from sklearn.model_selection import train_test_split

from xgboost import XGBRegressor

from sklearn import metrics

%matplotlib inline

from scipy import stats
```

loading the data from csv file to Pandas DataFrame

```
big_mart_data = pd.read_csv('Train.csv')
```

first 5 rows of the dataframe

```
big_mart_data.head()
```

number of data points & number of features

```
big_mart_data.shape
```

getting some information about the dataset

```
big_mart_data.info()
```

#Categorical Features:

```

#Item_Identifier
#Item_Fat_Content
#Item_Type
#Outlet_Identifier
#Outlet_Size
#Outlet_Location_Type
#Outlet_Type
# checking for missing values
big_mart_data.isnull().sum()
# mean value of "Item_Weight" column
big_mart_data['Item_Weight'].mean()
# filling the missing values in "Item_weight column" with "Mean" value
big_mart_data['Item_Weight'].fillna(big_mart_data['Item_Weight'].mean(),
inplace=True)
# mode of "Outlet_Size" column
big_mart_data['Outlet_Size'].mode()
# filling the missing values in "Outlet_Size" column with Mode
mode_of_Outlet_size = big_mart_data.pivot_table(values='Outlet_Size',
columns='Outlet_Type', aggfunc=(lambda x: x.mode()[0]))
miss_values = big_mart_data['Outlet_Size'].isnull()
print(miss_values)
big_mart_data.loc[miss_values, 'Outlet_Size'] =
big_mart_data.loc[miss_values, 'Outlet_Type'].apply(lambda x:
mode_of_Outlet_size[x])
big_mart_data.describe()
sns.set()
# Item_Weight distribution
plt.figure(figsize=(6,6))
sns.distplot(big_mart_data['Item_Weight'])
plt.show()
# Item Visibility distribution
plt.figure(figsize=(6,6))
sns.distplot(big_mart_data['Item_Visibility'])
plt.show()

```

```

# Item MRP distribution
plt.figure(figsize=(6,6))
sns.distplot(big_mart_data['Item_MRP'])
plt.show()

# Item_Outlet_Sales distribution
plt.figure(figsize=(6,6))
sns.distplot(big_mart_data['Item_Outlet_Sales'])
plt.show()

# Outlet_Establishment_Year column
plt.figure(figsize=(6,6))
sns.countplot(x='Outlet_Establishment_Year', data=big_mart_data)
plt.show()

# Item_Fat_Content column
plt.figure(figsize=(6,6))
sns.countplot(x='Item_Fat_Content', data=big_mart_data)
plt.show()

# Outlet_Size column
plt.figure(figsize=(6,6))
sns.countplot(x='Outlet_Size', data=big_mart_data)
plt.show()

big_mart_data.head()
big_mart_data['Item_Fat_Content'].value_counts()
big_mart_data.replace({'Item_Fat_Content': {'low fat':'Low Fat','LF':'Low Fat',
'reg':'Regular'}}), inplace=True)
big_mart_data['Item_Fat_Content'].value_counts()

```

Label Encoding

```

encoder = LabelEncoder()
big_mart_data['Item_Identifier'] =
encoder.fit_transform(big_mart_data['Item_Identifier'])

big_mart_data['Item_Fat_Content'] =
encoder.fit_transform(big_mart_data['Item_Fat_Content'])

```

```

big_mart_data['Item_Type'] = encoder.fit_transform(big_mart_data['Item_Type'])

big_mart_data['Outlet_Identifier'] =
encoder.fit_transform(big_mart_data['Outlet_Identifier'])

big_mart_data['Outlet_Size'] = encoder.fit_transform(big_mart_data['Outlet_Size'])

big_mart_data['Outlet_Location_Type'] =
encoder.fit_transform(big_mart_data['Outlet_Location_Type'])

big_mart_data['Outlet_Type'] = encoder.fit_transform(big_mart_data['Outlet_Type'])
big_mart_data.head()
X = big_mart_data.drop(columns='Item_Outlet_Sales', axis=1)
Y = big_mart_data['Item_Outlet_Sales']
print(X)
print(Y)

```

Splitting the data into Training data & Testing Data

```

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,
random_state=2)
print(X.shape, X_train.shape, X_test.shape, Y_test.shape)

```

Machine Learning Model Training

XGBoost Regressor

```

regressor = XGBRegressor()
regressor.fit(X_train, Y_train)
# prediction on training data
training_data_prediction = regressor.predict(X_train)
# R squared Value
r2_train = metrics.r2_score(Y_train, training_data_prediction)
print('R Squared value = ', r2_train)
# prediction on test data
test_data_prediction = regressor.predict(X_test)
# R squared Value

```

```

r2_test = metrics.r2_score(Y_test, test_data_prediction)
print('R Squared value = ', r2_test)

#RMSE

from sklearn.metrics import mean_squared_error

from math import sqrt

rmse_xgboost = sqrt(mean_squared_error(Y_test, test_data_prediction))

rmse_xgboost

```

Linear Regressor

```

new_data = data.copy()

#Independent Variables:

x = new_data.drop("item_outlet_sales", axis = 1)

#Depenedent Variables

y = new_data["item_outlet_sales"].values.reshape(-1,1)

#Splitting The data into Train and Test Dataset:

from sklearn.model_selection import train_test_split

x_train,x_test, y_train, y_test = train_test_split(x,y, test_size =0.20, random_state =
3)

#Applying Linear Regression Model

from sklearn.linear_model import LinearRegression

regressor =LinearRegression()

regressor.fit(x_train, y_train)

#Prediction

```

```

y_pred = regressor.predict(x_test)

#Accuracy of Model (Apply R2_score)

from sklearn.metrics import r2_score, mean_squared_error

r2_score(y_test, y_pred)


#Checking Root Mean Square error

from math import sqrt

rmse = sqrt(mean_squared_error(y_test, y_pred))

rmse

```

Ridge Regressor

```

#Ridge Regression

from sklearn.linear_model import Ridge

rr = Ridge(alpha = 0.009)

rr.fit(x_train, y_train)


rr_pred = rr.predict(x_test)

#Accuracy score check

r2_score(y_test, y_pred)


#RMSE

rmse_ridge = sqrt(mean_squared_error(y_test, rr_pred))

rmse_ridge

```

REFERENCES

- [1] Ching Wu Chu and Guoqiang Peter Zhang, "A comparative study of linear and nonlinear models for aggregate retails sales forecasting", *Int. Journal Production Economics*, vol. 86, pp. 217- 231, 2003.
- [2] Wang, Haoxiang. "Sustainable development and management in consumer electronics using soft computation." *Journal of Soft Computing Paradigm (JSCP)* 1, no. 01 (2019): 56.- 2. Suma, V., and Shavige Malleshwara Hills. "Data Mining based Prediction of D
- [3] Suma, V., and Shavige Malleshwara Hills. "Data Mining based Prediction of Demand in Indian Market for Refurbished Electronics." *Journal of Soft Computing Paradigm (JSCP)* 2, no. 02 (2020): 101- 110
- [4] Giuseppe Nunnari, Valeria Nunnari, "Forecasting Monthly Sales Retail Time Series: A Case Study", *Proc. of IEEE Conf. on Business Informatics (CBI)*, July 2017.
- [5] Maike Krause-Traudes et al. Spatial data mining for retail sales forecasting. Tech.rep. Fraunhofer-Institut Intelligente Analyse- und Informationssysteme (IAIS), 2008.
- [6] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.
- [7] Cerrada, M., & Aguilar, J. (2008). Reinforcement learning in system identification. In *Reinforcement Learning*. IntechOpen.
- [8] Welling, M. (2011). *A first encounter with Machine Learning*. Irvine, CA.: University of California, 12.
- [9] Learning, M. (1994). *Neural and Statistical Classification*. Editors D. Mitchie et. al, 350.
- [10] Mitchell, T. M. (1999). *Machine learning and data mining*. *Communications of the ACM*, 42(11), 30- 36.
- [11] Makridakis, S., Wheelwright, S.C., Hyndman, R.J.: *Forecasting methods and applications*. John wiley & sons (2008)
- [12] Kadam, H., Shevade, R., Ketkar, P. and Rajguru.: "A Forecast for Big Mart Sales Based on Random Forests and Multiple Linear Regression." (2018).
- [13] C. M. Wu, P. Patil and S. Gunaseelan: *Comparison of Different Machine Learning Algorithms for Multiple Regression on Black Friday Sales Data* (2018).
- [14] Das, P., Chaudhury: *Prediction of retail sales of footwear using feed forward and recurrent neural networks* (2018)

- [15] Das, P., Chaudhury, S.: Comparison of Different Machine Learning Algorithms for Multiple Regression on Black Friday Sales Data (2007)
- [16] G. Behera and N. Nain, "A Comparative Study of Big Mart Sales Prediction", pp. 1-13, 2019. [Accessed 10 October 2019].
- [17] S. Beheshti-Kashi, H. Karimi, K. Thoben and M. Lütjen, "A survey on retail sales forecasting and prediction in fashion markets", Systems Science & Control Engineering, vol. 3, no. 1, pp. 154-161, 2014. Available: 10.1080/21642583.2014.999389 [Accessed 27 January 2020].
- [18] A. Chandel, A. Dubey, S. Dhawale and M. Ghuge, "Sales Prediction System using Machine Learning", International Journal of Scientific Research and Engineering Development, vol. 2, no. 2, pp. 1-4, 2019. [Accessed 27 January 2020].
- [19] M. Wistuba, N. Schilling and L. Schmidt-Thieme, "Hyperparameter Optimization Machines", 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Montreal, QC, 2016, pp. 41- 50.
- [20] M. Wistuba, N. Schilling and L. Schmidt-Thieme, "Learning hyperparameter optimization initializations", 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Paris, 2015, pp. 1-10.