ANALYSING PERFORMANCE: A COMPARITIVE STUDY OF MACHINE LEARNING MODELS

A Thesis submitted
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

In

Applied Mathematics

by

Diksha Bhati
(2K22/MSCMAT/09)

Himani
(2K22/MSCMAT/17)

Under the supervision of

Prof. Anjana Gupta

Department of Applied Mathematics

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daultapur, Main Bawana Road, Delhi-42

May, 2024

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CANDIDATE’S DECLARATION

We, (Diksha Bhati) 2K22/MSCMAT/09 and (Himani) 2K22/MSCMAT/17 hereby certify that the work which is being presented in the thesis entitled “Analysing Performance: A Comparative Study of Machine Learning Models” in partial fulfillment of the requirement for the award of the Degree of Master of Science, submitted in the Department of Applied Mathematics, Delhi Technological University is an authentic record of my own work carried out during the period from August 2023 to April 2024 under the supervision of Professor Anjana Gupta.

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Candidate’s Signature

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.

Signature of Supervisor **Signature of External Examiner**

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CERTIFICATE BY THE SUPERVISOR

Certified that Diksha Bhati (2K22/MSCMAT/09) and Himani (2K22/MSCMAT/17) has carried out their search work presented in this thesis entitled “Analysing Performance: A Comparative Study of Machine Learning Models” for the award of Master of Science from Department of Applied Mathematics, Delhi Technological University, Delhi, under my supervision. The thesis embodies results of original work, and studies are carried out by student themselves and content of the thesis do not form the basis for the award of any other degree to the candidates or to anybody else from this or any other University/Institution.

Place: Delhi
Date: June, 2024

Prof. ANJANA GUPTA
SUPERVISOR
DEPARTMENT OF APPLIED MATHEMATICS
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-
110042

ANALYSING PERFORMANCE: A COMPARATIVE STUDY OF MACHINE LEARNING MODELS

Diksha Bhati and Himani

ABSTRACT

Sales forecasting is a fundamental company technique that influences inventory management, revenue estimation, and investment decisions. This study compares machine learning algorithms for improving sales projections, taking into account the financial implications of fluctuating sales seasons. Failure to foresee sales patterns can lead to significant losses, particularly for hug firms such as Walmart.

The study makes use of a thoroughly pre-processed and merged dataset that includes features such as store details, department temperature, fuel price, markdowns, and the consumer price index (CPI), as well as five markdown features. Data uniformity is emphasized to assure model effectiveness by assigning equal weight to each feature and supporting a variety of techniques. Five machine learning models are used for comparative study. Random forest outperforms the others, obtaining a sales forecasting. The study underlines the importance of data standardization and preprocessing in model performance and offers guidance on model selection for sales forecasting.

To summarize, this study contributes to the field of sales forecasting and offers practical advice for firms looking for effective machine learning models for strategic decision-making. The findings provide useful guidelines for using predictive analytics in the retail industry, emphasizing the necessity of recognizing and adjusting to sales changes for long-term economic success.

ACKNOWLEDGEMENTS

At the outset of this report, we extend our heartfelt appreciation to all individuals who have supported us in completing this dissertation. Without their proactive direction, assistance, collaboration, and support, we could not have advanced toward achieving the desired outcomes. Prof. Anjana Gupta provided diligent assistance and support that enabled us to complete our dissertation, for which we are eternally grateful. We express our sincere appreciation to each other for working together to complete this project while preserving our individuality. We are thankful that Delhi Technological University provided this opportunity to us. We additionally express our sincere gratitude and respect to our parents as well as other family members, who have always provided us with both material and moral support. Finally, but just as importantly, we would like to express our heartfelt gratitude to all of our friends who supported us in any way during this effort. This quick acknowledgement does not imply a lack of gratitude for anything.

Thanking You

DIKSHA BHATI
HIMANI

TABLE OF CONTENTS

List of Figures

List of Tables

List of Abbreviations

ML-Machine Learning

MAE- Mean Absolute Error

MSE-Mean Square Error

RMSE- Root Mean Square

SVM-Support Vector Machine

KNN- K-nearest neighbor

CHAPTER 1

INTRODUCTION

1.1 Machine Learning-

The globe has been dominated by the Internet in recent years, and there has been a notable growth in the flow of data through the internet. There are huge chunks of organized and unorganized data available in the market. Maintaining and extracting information from the data becomes more difficult as data flow increases day by day. Data enables a number of businesses to use the data to create decisions that can be concluded and represented by enterprises.

Finding the required information from the given data, which is haphazardly or partially arranged, is an extremely hectic process. Data must be organized by machines, and those machines must be intelligent enough to extract pertinent information from large amounts of data; otherwise, the process would take years and be useless. In order for machines to continue learning from their prior experiences, we must teach them.

In recent years, artificial intelligence has reached new heights. Specifically, machine learning, deep learning, and natural language processing have gained popularity.

Machine learning works with computers and other devices, enabling them to learn without the need for external programming. Machine learning algorithms employ historical data as input to generate new output data. The goal is to enable machines to learn from past experience, analyse data, spot patterns, and make decisions without the need for human intervention.

1.2 Types of Machine Learning

1.2.1 Supervised Learning

This type of learning uses labelled data for model training. In this learning, the training data consists of dependent and independent variables, and the algorithm learns to associate inputs with the correct outputs by minimizing the error between its predictions and the true labels. This learning is frequently employed for tasks like classification, regression. For instance, it can be utilized to determine whether emails are spam or not.

It is widely used in Object Recognition, Speech Recognition, Spam Detection, Image Classification, Market Prediction and much more.

1.2.2 Unsupervised Learning

This type of learning uses unlabeled data for model training. In this learning, the algorithm seeks to discover patterns and structures in the data without predefined labels. Clustering, combining similar data points, and dimensionality reduction, which requires to reduce the number of features while maintaining the dataset's essential information, represent typical learning tasks.

This approach is commonly applied in exploratory data analysis, anomaly detection, and market segmentation.

1.2.3 Reinforcement Learning

One of the lesser-known aspects of Machine Learning is the concept of reinforcement learning, in which a subject connects with the outside world and maximizes the reward in order to learn how to make decisions.

Reinforcement learning focuses on identifying the optimal behavior in a given scenario to achieve the highest benefits. Unlike supervised or unsupervised machine learning, reinforcement learning does not rely on pre-existing data; instead, it learns from the outcomes of its actions, often through trial and error. This approach is commonly applied in areas such as robotic control, factory optimization, and video game playing.

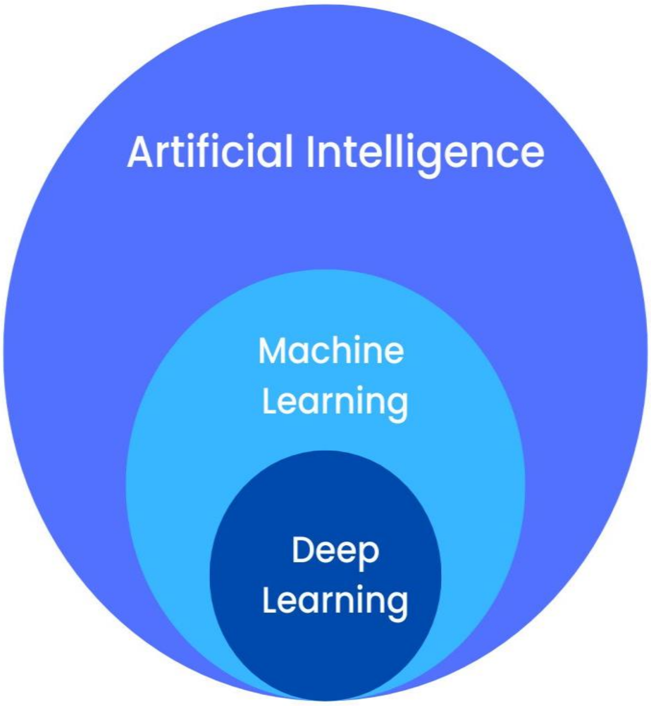


Fig 1.1 -Artificial Intelligence vs Machine Learning vs Deep Learning

CHAPTER – 2
LITREATURE REVIEW

- In [1] Bohdan M. Pavlyshenko stated this research, they investigate the use of machine learning models to predictive analytics for sales. This thesis states that primary goal is to examine key policies for using machine learning for sales forecasting. When a new product or shop is introduced and there is little previous data available for a particular sales time series, this impact might be utilized to forecast sales. A method for constructing an ensemble of single models for regression has been examined. The outcomes demonstrate the use of stacking methods.

- In [2] Singh Manpreet, Bhawick Ghutla, Reuben Lilo Jnr, Aesaan FS Mohammed, and Mahmood A. Rashid tells us the main objective of this study is to apply big data analytics to analyse sales data from Walmart. The writers talk about the difficulties in evaluating vast volumes of sales data as well as the possible advantages of big data analytics. Additionally, they provide the findings of their investigation, which included creating sales forecasting models and spotting patterns and trends in sales data.

- In [3] J Saran Prakash, Dr. Swati Sah states that the purpose of this study is to examine big data analytics and machine learning methods used in classic and contemporary approaches to sales prediction. While the current technique uses machine learning algorithms like random forests, decision trees and neural networks, the conventional approach uses statistical approaches and time-series analysis to anticipate sales. They assess each approach's possible advantages and disadvantages and compare the two methods' accuracy, speed, and scalability. According to their findings, machine learning algorithms achieve much greater prediction accuracy than conventional statistical approaches, indicating that the current methodology performs more accurately than the traditional way. Furthermore, compared to the conventional technique, the current approach is faster and more scalable, allowing organizations to examine greater datasets and make predictions more quickly. But the contemporary strategy could be more difficult to implement and need more computer power than the conventional method.

- In [4] Asmita Manna, Kavita Kolpe, Aniket Mhalungekr, Sainath Pattewar, Pushpak Kaloge, Ruturaj Patil discovered in the pharmaceutical industry that the limiting medication waste because of expiration dates is essential. Currently, the majority of drug dealers and pharmacists use their personal sales expertise to manually forecast future drug sales. But by forecasting medicine sales based on historical sales data, artificial intelligence and machine learning can be quite helpful in this situation. To forecast future sales and assess the accuracy of several algorithms on a few particular types of the most popular pharmaceuticals worldwide, this study uses various machine-learning methods on sales data. The dataset that was utilized included sales data from a variety of medications, including antipyretics and antihistamines. Their results showed that The XGBoost Model worked successfully compared to the other models for the prediction.

- In [5] Rao Faizan Ali, Amgad Muneer, Ahmed Almaghthawi, Amal Alghamdi, Suliman Mohamed Fati, Ebrahim Abdulwasea, Abdullah Ghaleb discovered that for many merchants, the primary problem is fluctuations in sales over time. In an effort to solve this issue, they compare the historical sales data of various retailers in an effort to forecast the sales. Predicting the monthly sales value is important for the research while tackling this topic. Clearing out the missing data and doing appropriate feature engineering are also crucial steps in order to better understand the features before implementing them. Out of these machine-learning approaches used in here, the random forest predictor performed better, according to the experimental data, than ridge regression; linear regression; and decision trees. Root mean square error (RMSE) has been used to assess the performance of the suggested models.

- In [6] Sathyanarayana S, Apeksha C, Chethana S, Chinmayee H C, Abhishree G L states that software programs can anticipate outcomes more accurately using Machine Learning, many algorithms eliminate the need for explicit programming. In order to forecast the sales of various qualities and item types as well as to comprehend the influences of various elements on the sales of the products, this article has examined the instance of Shopping Mart, a one-stop shopping centre. Taking into account several facets of the dataset gathered for Shopping-Mart and the construction process used to build a model, extremely precise outcomes are produced.

- In [7] Gopal Behera and Neeta Nain they offer a predictive model for estimating sales of a firm like Shopping Mart using the Xgboost approach. They find that the model performs better than current models. A thorough explanation is also provided of a comparison of the model's performance measures with those of other models.

- In [8] Sushila Ratre & Jyotsna Jayaraj in their research compares several machine learning models that are utilized for sales predictive analytics. A crucial component of contemporary corporate intelligence, sales forecast helps with staff scheduling and inventory management, which in turn improves sales and customer loyalty. The drugstore chain Rossmann is the subject of the dataset that was chosen. By managing outliers and missing data, feature engineering aids in data cleaning and the understanding of the relative value of distinct characteristics. The dataset is found to have seasonal tendencies and to be historical in character. Three models have been selected for comparison: XGBoost, Facebook's Prophet, and ARIMA. Because of the circumstances, the forecast accuracy is determined by the Root Mean Square Error (RMSE). The ARIMA Model performs better than the others, as seen by its lowest value of 739.06 in the findings.

- In [9] Younes Ledmaouia∗, Adila El Maghraouib, Mohamed El Aroussia, Rachid Saadanea, Ahmed Chebakb, Abdellah Chehric describes a comprehensive and comparative study of solar energy production forecasting in Morocco using six machine learning (ML) algorithms: Support Vector Regression (SVR), Artificial Neural Network (ANN), Decision Tree (DT), Random Forest (RF), Generalized Additive Model (GAM), and Extreme Gradient Boosting (XGBOOST). Following testing and training, the models were assessed. Four metrics were employed in this study to evaluate the performance of the model. The ANN model is the most effective predictive model for energy forecasting in scenarios, based on the models' results comparable to the one in question, with the lowest RMSE, MSAE, and greatest *R*-squared values—all of which the ANN model recognizes as critical performance metrics.

- In [10] Jiaming Chen focuses on recommendations for lung cancer prevention and lung cancer prediction. They will evaluate many kinds of machine learning prediction models and algorithms in order to determine which class of algorithmic models is most suited for predicting lung cancer. The random forest algorithm model, which represents the ensemble learning algorithm model, is the best class of prediction algorithm models, according to the results of the experimental comparison, secondary and tertiary indicator results, and ROC curves produced by the confusion matrix.

CHAPTER 3
DATA OVERVIEW

There are numerous seasons when sales are noticeably above or below average. The corporation may suffer excessive financial losses if it is unaware of these seasons. Forecasting sales is one of a business's most important strategies.

The company may allocate inventory, project income, and decide whether to make additional investments with the help of sales forecasting. Knowing what to expect from future sales also helps because meeting goals set at the start of the season can boost stock values and investor sentiment. Conversely, failing to meet the anticipated goal could seriously hurt stock prices. And it will be a major issue, particularly for Walmart given its size. In our dataset, we have three CSV files. We did the preprocessing, merged those files, and got the final dataset. The first dataset contains five features. The features are ‘Store’, ‘Dept’, 'Date', 'Weekly Sales', and 'Isholiday'. This dataset does not have any null values and contains about four lakh twenty-one thousand five hundred seventy records. There are around 45 unique stores in the data and 59 unique departments.

The second dataset contains information about the stores. Basically, we have three features. The features are ‘store’, ‘type’ and ‘size’. Stores are numbered from 1 to 45, and we have 3 types of stores: A, B, and C.

The final dataset talks about the features that contain data related to store, department temperature, fuel price, markdown, and consumer price index. We have five markdown features: 'Markdown1', 'Markdown2', 'Markdown3', 'Markdown4', and 'Markdown5'. Markdown refers to the anonymised data associated with Walmart's promotional markdowns.

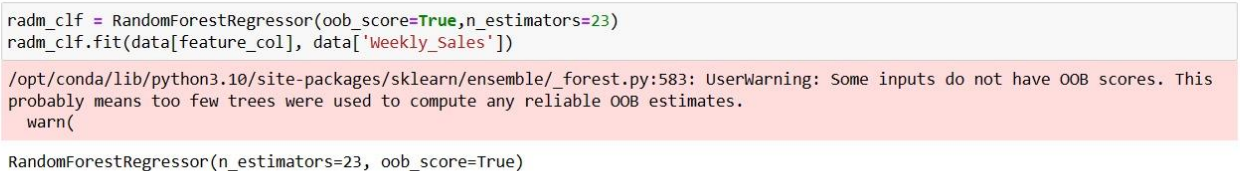
3.1 DATA PREPROCESSING

Data preprocessing plays a crucial role in building a model. So, we have pre- processed the data before building the model by fetching information regarding the data. So basically, we have 3 datasets, i.e., one dataset has dates and weekly sales features, including other 3 features in which we do not have any null values present; the other dataset is about the store type and store sizes and contains 0 null values; and the last dataset is about the features of the store; it contains features like date, temperature, fuel prize, markdowns, holidays, CPI, and unemployment. This dataset contains null values. We checked the datatype of every feature. We have replaced those null values with the median of the respective features of the CPI and unemployment features. We then replace all the19negative and null values in Markdowns features with 0 values. After this, we joined two data sets and stored data on the ‘Store’ feature. After that, we merge the data with feature data on the ‘Store’ and ‘Date’ features. We then calculated the ‘Total Markdown’ features and dropped the Markdown features. Next, we converted the ‘Isholiday’ Boolean datatype to an int datatype. We have ‘Store’, ‘Dept’, and ‘Type’ columns, which are categorical columns. We converted these categorical columns using Get Dummies in the Pandas library. Our features have different ranges. We need to standardize the data so that we can give equal weight to each of the features, and standardization is also needed in distance-based algorithms and in algorithms that use gradient descent techniques to optimize the model. We created three new features from the ‘Date’ feature by extracting date, month, and year from the date. We used the minmax scaling technique to normalize the dataset.

The MinMax Scaler minimize the given data inside the specified range, which is typically [0,1]. Data is transformed by scaling characteristics to a specified 17-range
It preserves the original distribution's structure while scaling the numbers to a predetermined range. The following is the mathematical formula used in minmax scaling -

$$\alpha\ \_std = (\alpha - \alpha.min) / (\alpha.max - \alpha.min)\ \alpha$$
$$\_scaled = \alpha\ \_std * (max - min) + min$$

As we have 145 features, we have to use the feature elimination technique in order to decrease the number of features so that we don’t face the curse of dimensionality. For feature importance, we used Random Forest Regressor and then found their importance and sorted them in descending order. After that, we took 27 features that are most important for our model’s accuracy.

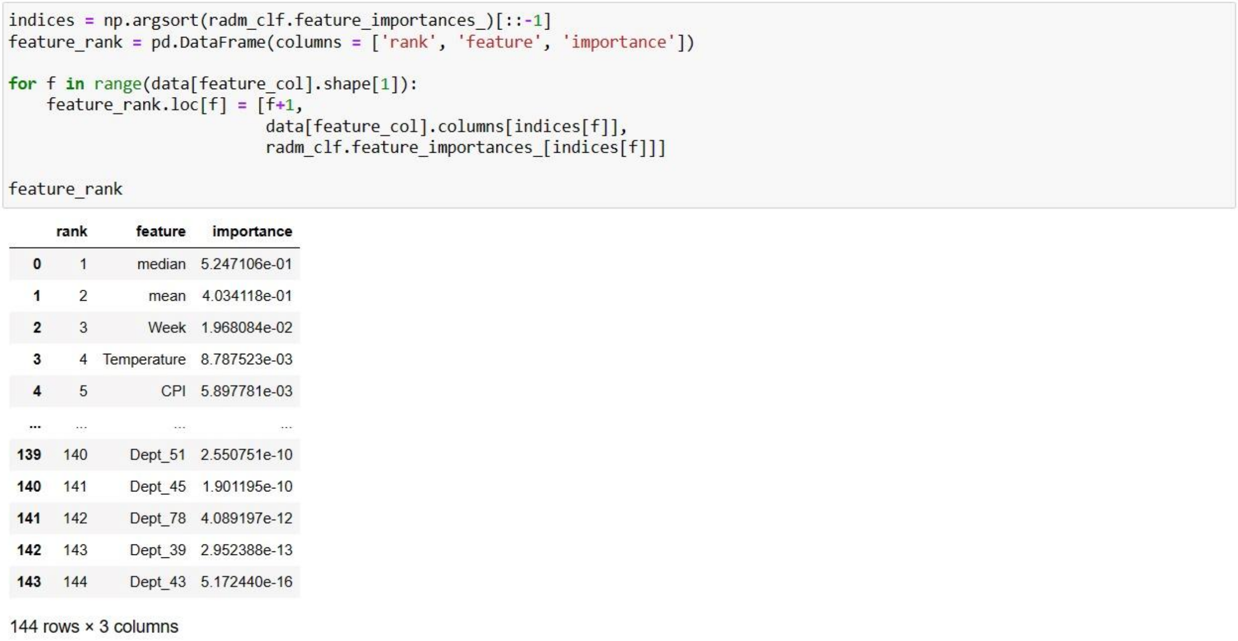


```
radm_clf = RandomForestRegressor(oob_score=True,n_estimators=23)
radm_clf.fit(data[feature_col], data['Weekly_Sales'])
```

```
/opt/conda/lib/python3.10/site-packages/sklearn/ensemble/_forest.py:583: UserWarning: Some inputs do not have OOB scores. This
probably means too few trees were used to compute any reliable OOB estimates.
  warn(
```

```
RandomForestRegressor(n_estimators=23, oob_score=True)
```

Fig 3.1 – Feature Elimination

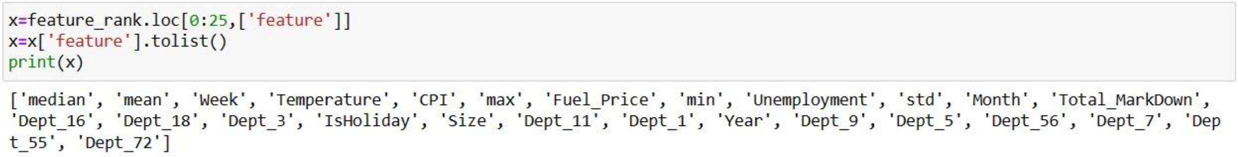


```
indices = np.argsort(radm_clf.feature_importances_)[::-1]
feature_rank = pd.DataFrame(columns = ['rank', 'feature', 'importance'])

for f in range(data[feature_col].shape[1]):
    feature_rank.loc[f] = [f+1,
                           data[feature_col].columns[indices[f]],
                           radm_clf.feature_importances_[indices[f]]]

feature_rank
```

	rank	feature	importance
0	1	median	5.247106e-01
1	2	mean	4.034118e-01
2	3	Week	1.968084e-02
3	4	Temperature	8.787523e-03
4	5	CPI	5.897781e-03
...	...	...	...
139	140	Dept_51	2.550751e-10
140	141	Dept_45	1.901195e-10
141	142	Dept_78	4.089197e-12
142	143	Dept_39	2.952388e-13
143	144	Dept_43	5.172440e-16

144 rows × 3 columns

Fig 3.2– Feature Elimination



```
x=feature_rank.loc[0:25,['feature']]
x=x['feature'].tolist()
print(x)
```

```
['median', 'mean', 'Week', 'Temperature', 'CPI', 'max', 'Fuel_Price', 'min', 'Unemployment', 'std', 'Month', 'Total_MarkDown',
'Dept_16', 'Dept_18', 'Dept_3', 'IsHoliday', 'Size', 'Dept_11', 'Dept_1', 'Year', 'Dept_9', 'Dept_5', 'Dept_56', 'Dept_7', 'Dep
t_55', 'Dept_72']
```

Fig 3.3– Selected Features from the Dataset

3.2 EXPLORATORY DATA ANALYSIS

From the exploratory data analysis, we got some findings. Some departments and stores experience higher sales during specific seasons, such as Thanksgiving. It suggests that certain events or holidays have a significant impact on sales. Based on their sizes, stores are divided into three kinds (H, I and J).

Nearly all the retailers are larger than 150,000 and fall into category A. Sales vary based on store type. As expected, holiday average sales are higher than on regular dates, Black Friday, Thanksgiving, Christmas, and the 22nd week of the year are somewhat the top four sales periods.

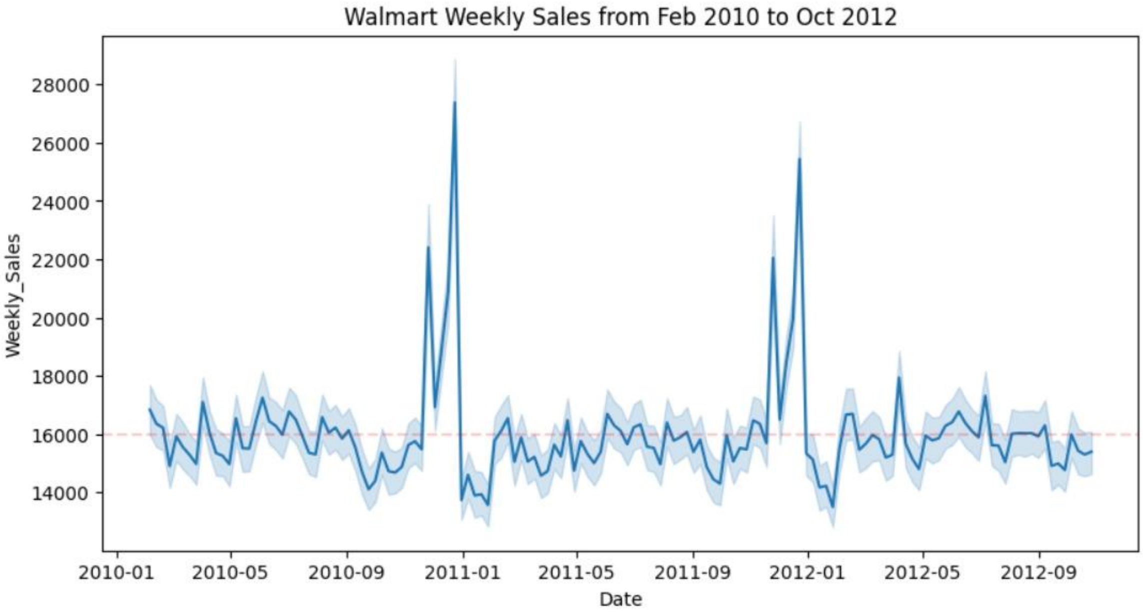


Fig 3.4 Weekly Sales

- We can see the increase in sales in November (11) and December (12)
- June (6) also has slightly higher sales.
- January (1) sales are lowest, it can be due to people avoiding shopping just after the holiday season

Although Christmas falls at the end of the year, people typically do more shopping during the 51st week than in the final days of December. January sales are considerably lower compared to other months, likely because of the high sales volumes seen in November and December. Consumers may be more conservative in their spending after the holiday season. Weekly sales are not strongly influenced by consumer price index (CPI), temperature, unemployment rate, or fuel prices.

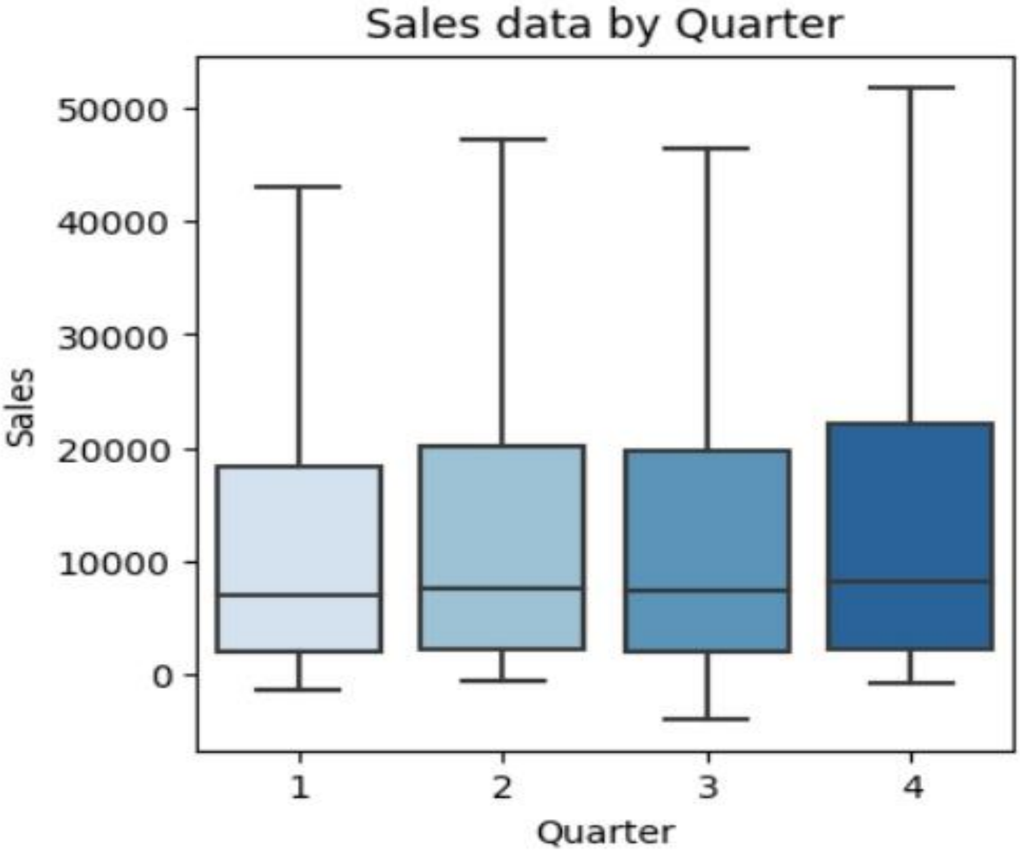


Fig 3.5- Q4 has the highest sales due to holiday season

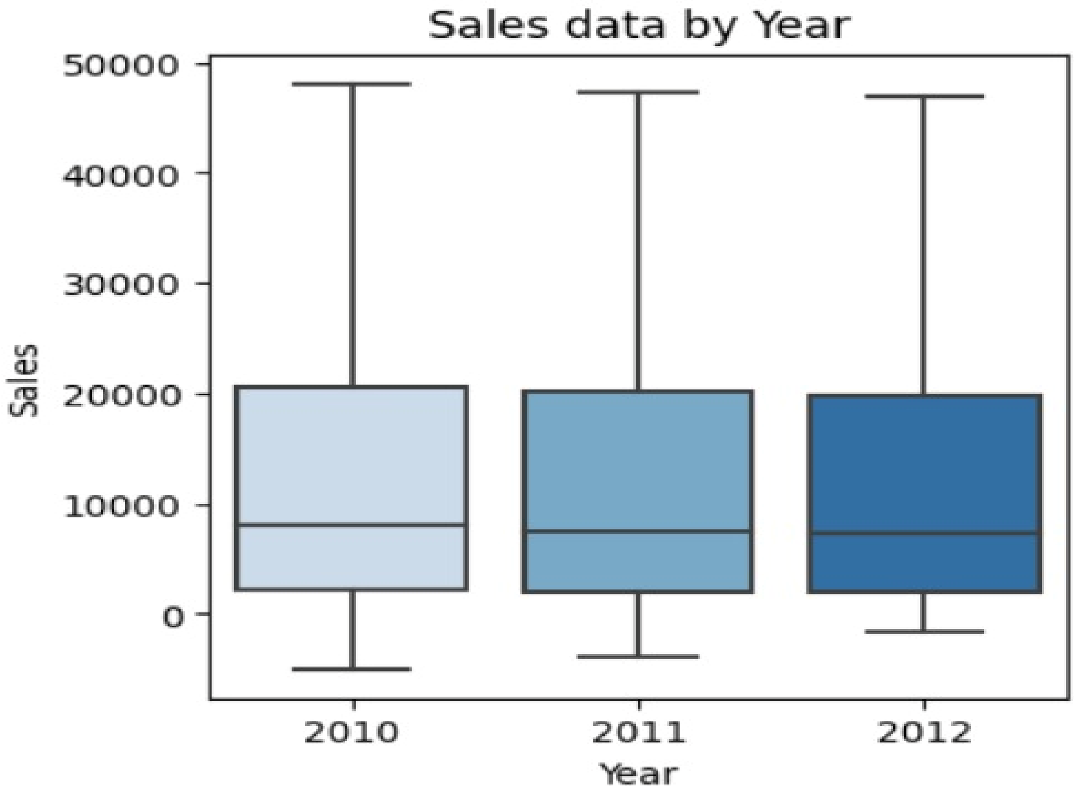


Fig 3.6 – Sales Data by Year

We see a decrease in Sales from 2010 to 2012, It can happen due to the following reasons:
We don’t have sales of Nov and Dec in 2012
We also don’t have sales of Jan in 2010 (which might make the average a little higher than 2011)

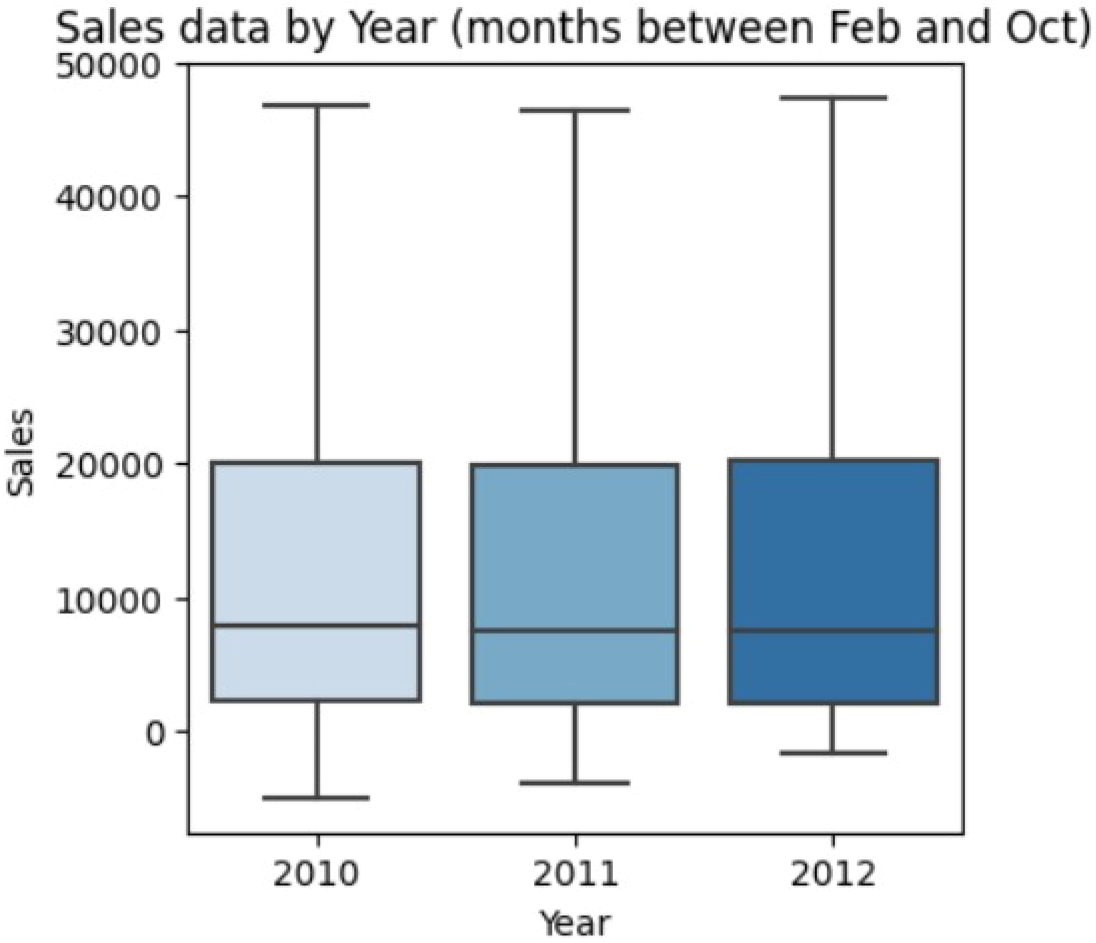


Fig 3.7 -Sales data by Year (b/w Feb and Oct)

Taking data of Months Feb-Oct as they are available for all years

We can see the sales is not decreasing over the years

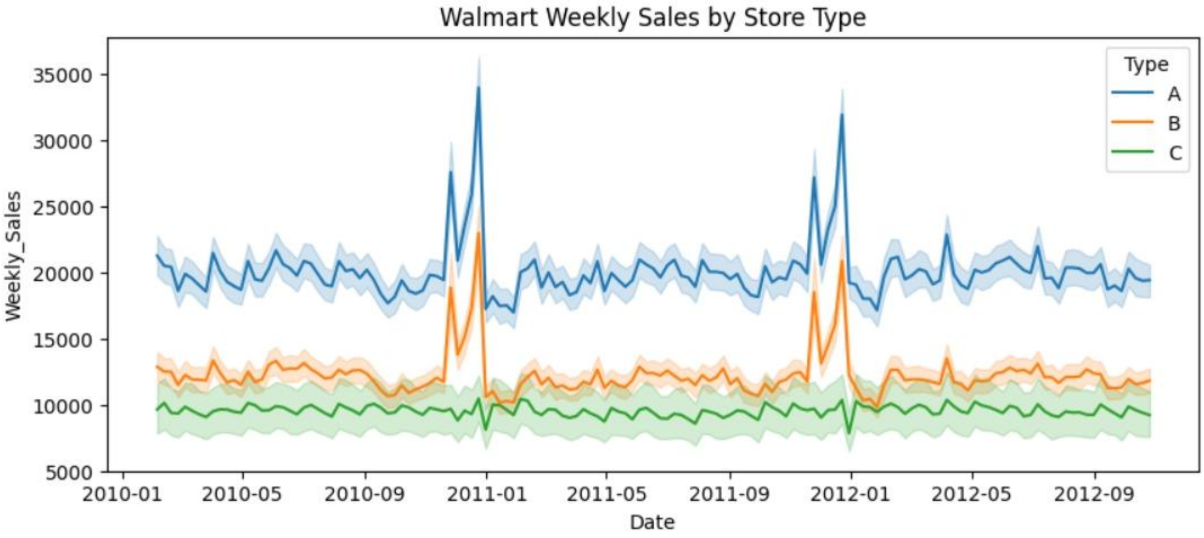


Fig 3.8- Sales by store type

Bigger stores (A) have higher sales compared to smaller stores (B)

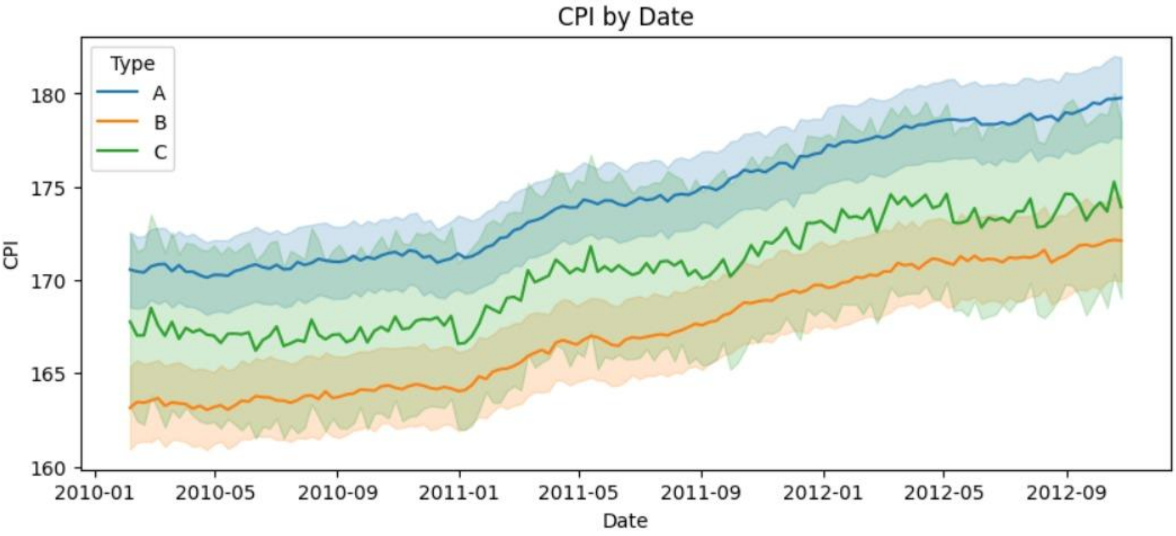
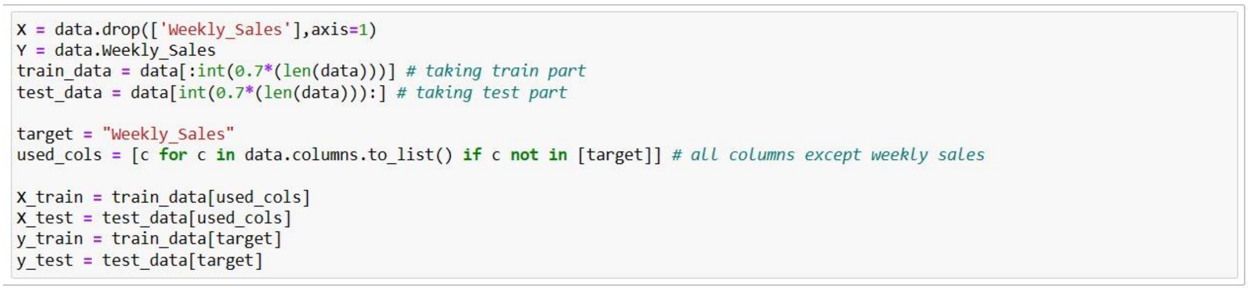


Fig 3.9- Overall CPI (Consumer Price Index) is increase over time
Type A stores have highest CPI index
Type C stores have higher CPI index than Type B stores

CHAPTER-4

MODEL BUILDING

We developed five machine learning models for comparison. Training a machine learning model involves instructing an algorithm to identify patterns and make predictions based on the data. We will start by dividing the data into 70% for training and 30% for testing.



```
X = data.drop(['Weekly_Sales'],axis=1)
Y = data.Weekly_Sales
train_data = data[:int(0.7*(len(data)))] # taking train part
test_data = data[int(0.7*(len(data))):] # taking test part

target = "Weekly_Sales"
used_cols = [c for c in data.columns.to_list() if c not in [target]] # all columns except weekly sales

X_train = train_data[used_cols]
X_test = test_data[used_cols]
y_train = train_data[target]
y_test = test_data[target]
```

Fig 4.1- Dividing the dataset into Training and Testing

Now, required models are trained on the training set, where it learns to map inputs to outputs.

The model's performance is evaluated using metrics like **MAE, MSE, R^2** and **RMSE.**

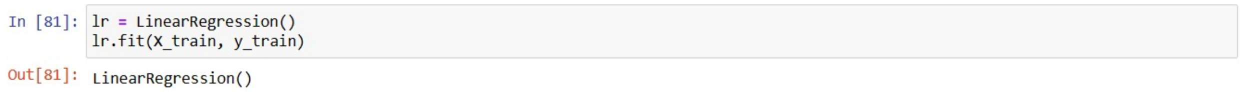
4.1 Linear Regression Model

A basic statistical and machine learning method for simulating the connection between an output (the dependent-variable) and more inputs (the independent-variables) is called linear regression. The goal of linear regression is to identify the best-fitting linear relationship between the variables. A basic linear regression model with a single independent variable may be expressed mathematically as follows:

$$y=mx+b$$

Here:

y-- dependent variable (output),
x --independent variable (input),
m -- slope of the line (coefficient),
b -- the y-intercept.

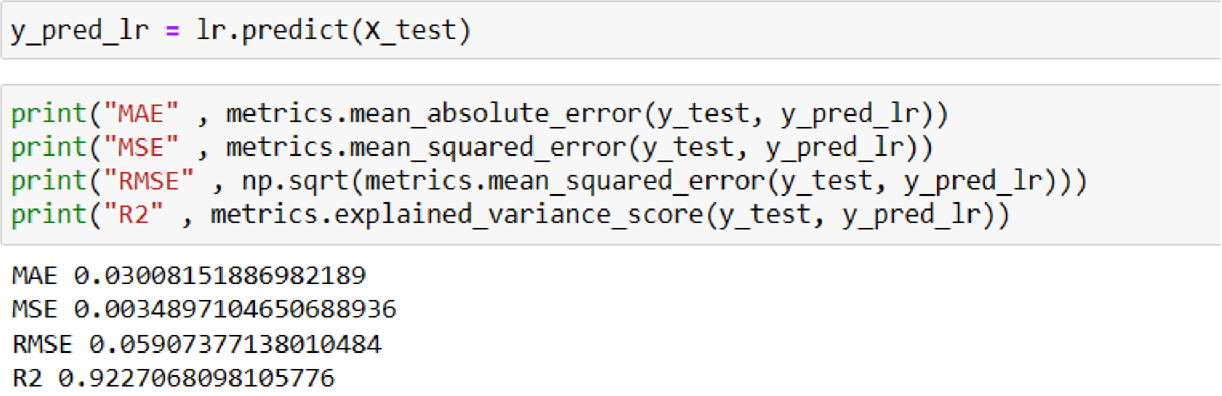


```
In [81]: lr = LinearRegression()
         lr.fit(X_train, y_train)

Out[81]: LinearRegression()
```

Fig 4.2 - Training Linear Regression

In linear regression, the aim is to determine the values of **m** and **b** that minimize the sum of squared differences between the observed and predicted values.

By linear regression model we got **0.029 MAE** on the test dataset. We also calculated **RMSE** which we got **0.53** and **R^2** score we got **0.93**. Below is the graph of actual and predicted values using linear regression model.

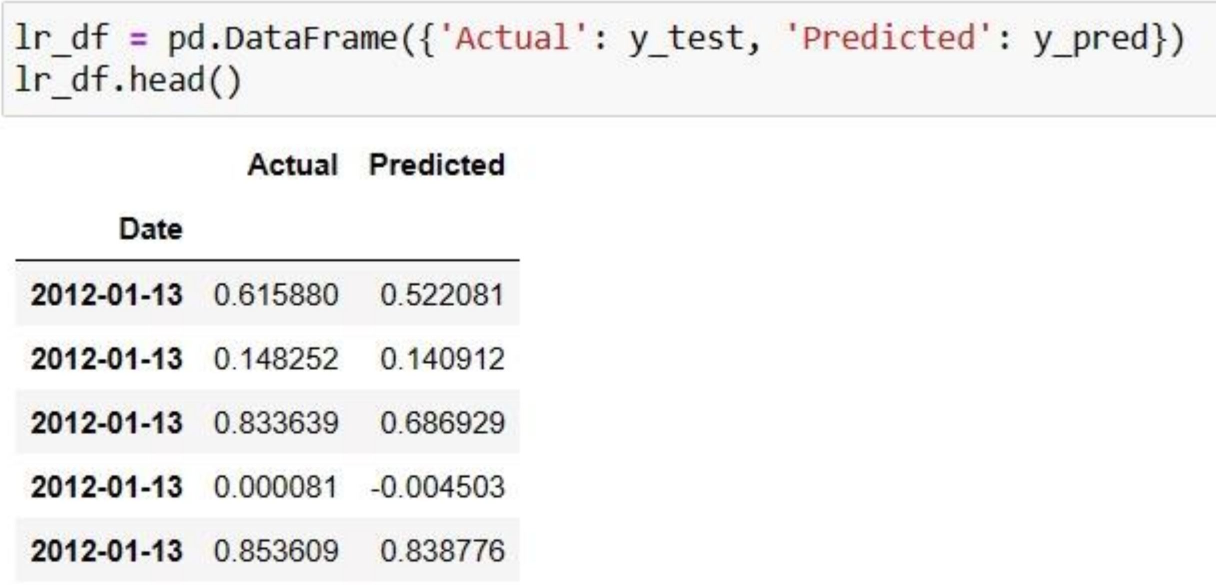


```
y_pred_lr = lr.predict(X_test)
```

```
print("MAE" , metrics.mean_absolute_error(y_test, y_pred_lr))
print("MSE" , metrics.mean_squared_error(y_test, y_pred_lr))
print("RMSE" , np.sqrt(metrics.mean_squared_error(y_test, y_pred_lr)))
print("R2" , metrics.explained_variance_score(y_test, y_pred_lr))
```

```
MAE 0.03008151886982189
MSE 0.0034897104650688936
RMSE 0.05907377138010484
R2 0.9227068098105776
```

Fig 4.3- MAE, MSE, RMSE and R^2 score in Linear Regression



```
lr_df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
lr_df.head()
```

Date	Actual	Predicted
2012-01-13	0.615880	0.522081
2012-01-13	0.148252	0.140912
2012-01-13	0.833639	0.686929
2012-01-13	0.000081	-0.004503
2012-01-13	0.853609	0.838776

Fig 4.4- Actual v/s Predicted value in Linear Regression

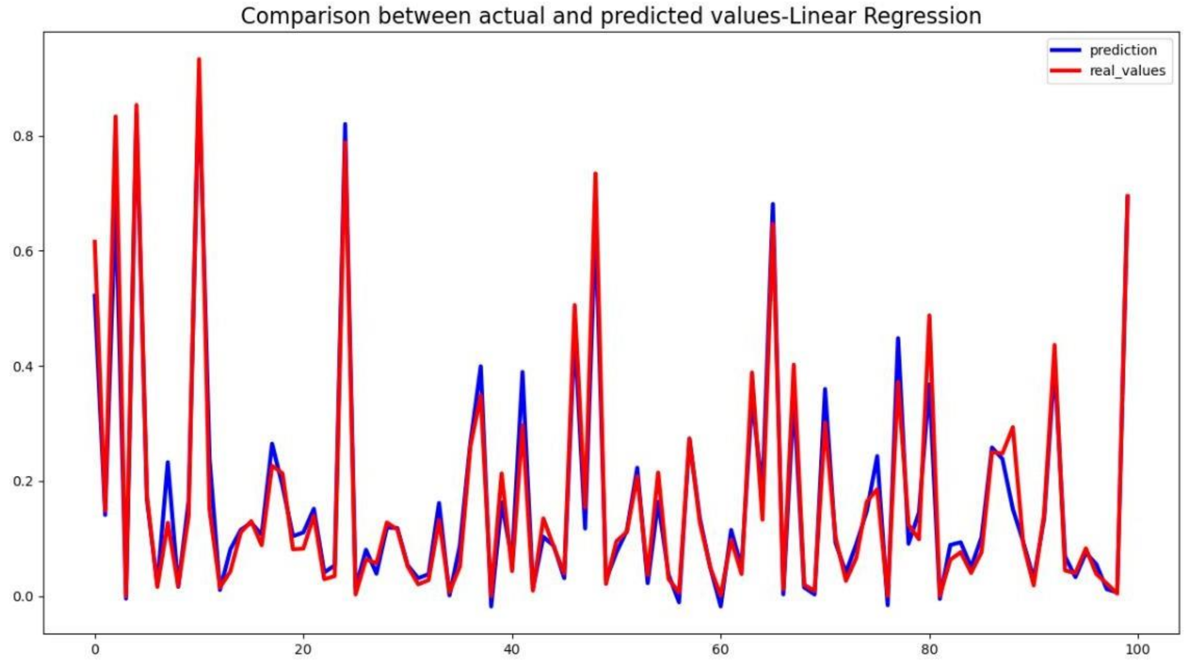


Fig 4.5- Line Graph of Actual v/s Predicted in Linear Regression

4.2 Random Forest Regressor

The second model utilized is the Random Forest Regressor, which employs ensemble learning by aggregating predictions from numerous decision trees to enhance predictive accuracy and mitigate overfitting. This algorithm operates based on key mathematical principles:--

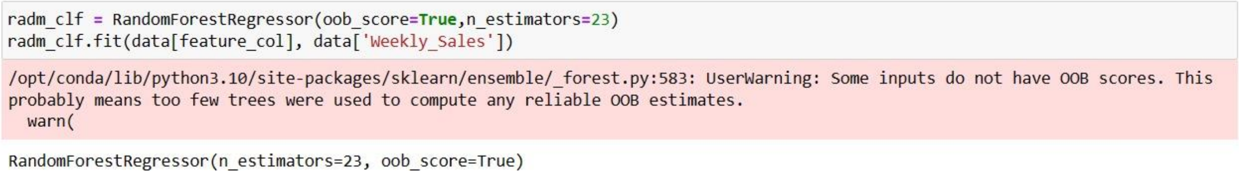
Bagging using Bootstrap Aggregation: By training each tree on a randomly selected portion of the training dataset, Random Forest creates several decision trees. By using a technique called bagging, the trees become more diverse.

The Randomization of Features: Every decision tree is built with a random subset of characteristics at each split, in addition to utilizing various subsets of the data. The resilience of the model is increased by this randomization, which also helps decorrelate the trees.

1. **Voting or Averaging:** Either voting (for classification) or averaging (for regression) is used to integrate the predictions from separate trees to determine the Random Forest's ultimate prediction. When compared to individual trees, this ensemble approach typically produces results that are more reliable and accurate.

2. **Out-of-Bag (OOB) Error:** The algorithm estimates the performance of the model using the out-of-bag samples, which are not part of each tree's training subset. In doing so, the correctness of the model can be objectively evaluated without requiring a different validation set.

3. **Split and Tree Growth Criteria:** Recursively dividing nodes according to standards similar to how mean squared error reduction is used for regression or Gini impurity for classification, the expansion of each decision tree in the forest is enabled.



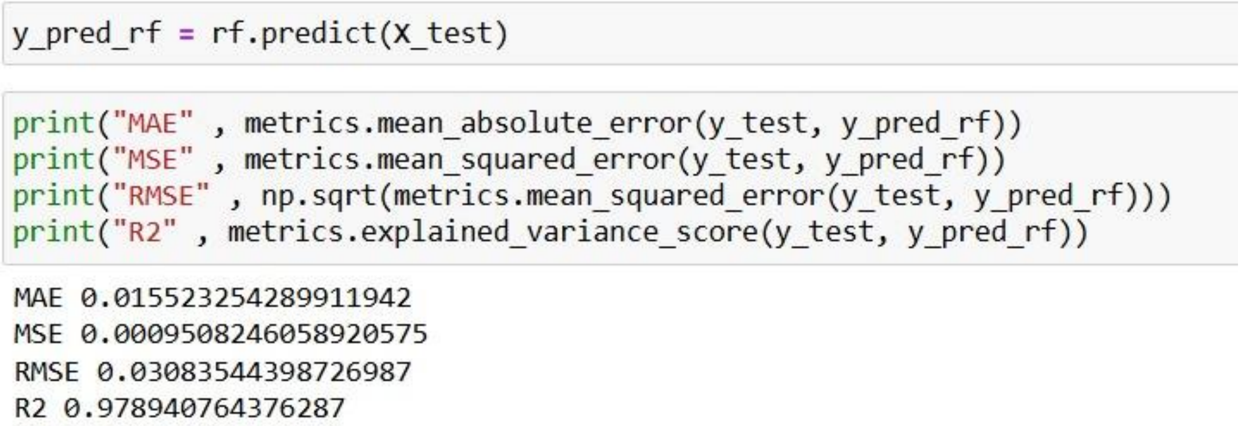
```
radm_clf = RandomForestRegressor(oob_score=True,n_estimators=23)
radm_clf.fit(data[feature_col], data['Weekly_Sales'])
```

```
/opt/conda/lib/python3.10/site-packages/sklearn/ensemble/_forest.py:583: UserWarning: Some inputs do not have OOB scores. This
probably means too few trees were used to compute any reliable OOB estimates.
  warn(
```

```
RandomForestRegressor(n_estimators=23, oob_score=True)
```

Fig 4.6- Training Random Forest

We used Random Forest regressor model because it is ensemble model and solves overfitting problem. We used **n_estimators=150** and **min_samples_split**
= 4 and got **0.020 MAE**. We also calculated **RMSE which we got 0.38** and **R^2 score we got 0.96.** Below is the graph of actual v/s predicted values by Random Forest regressor.

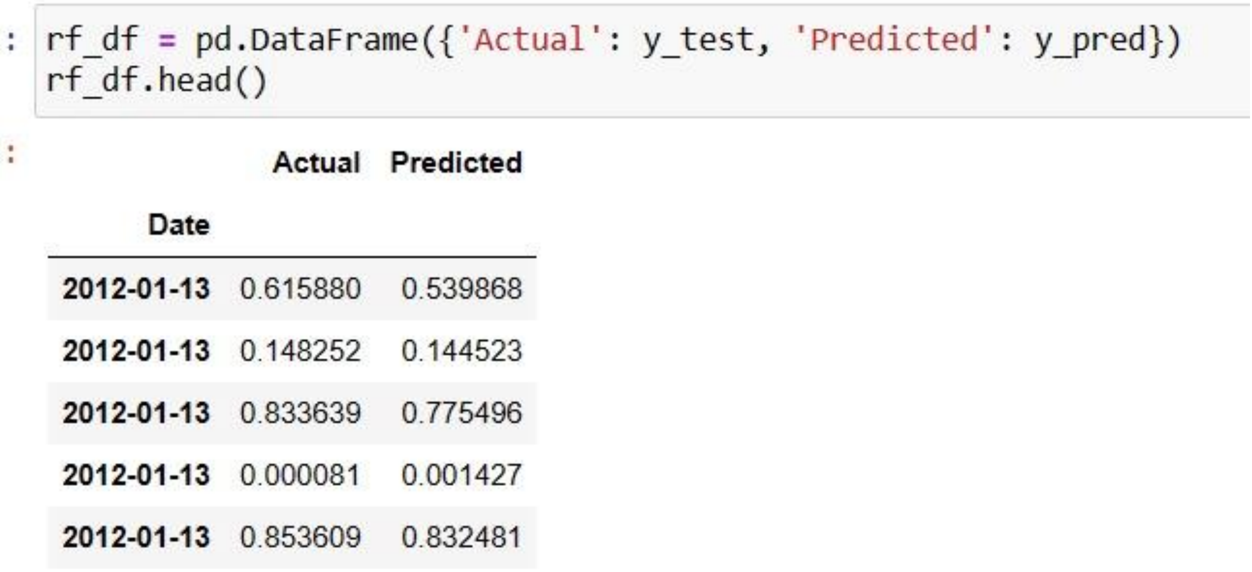


```
y_pred_rf = rf.predict(X_test)
```

```
print("MAE" , metrics.mean_absolute_error(y_test, y_pred_rf))
print("MSE" , metrics.mean_squared_error(y_test, y_pred_rf))
print("RMSE" , np.sqrt(metrics.mean_squared_error(y_test, y_pred_rf)))
print("R2" , metrics.explained_variance_score(y_test, y_pred_rf))
```

```
MAE 0.015523254289911942
MSE 0.0009508246058920575
RMSE 0.03083544398726987
R2 0.978940764376287
```

Fig 4.7-MAE, MSE, RMSE and R^2 score in Random Forest



```
rf_df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
rf_df.head()
```

	Actual	Predicted
Date		
2012-01-13	0.615880	0.539868
2012-01-13	0.148252	0.144523
2012-01-13	0.833639	0.775496
2012-01-13	0.000081	0.001427
2012-01-13	0.853609	0.832481

Fig 4.8- Actual v/s Predicted in Random Forest

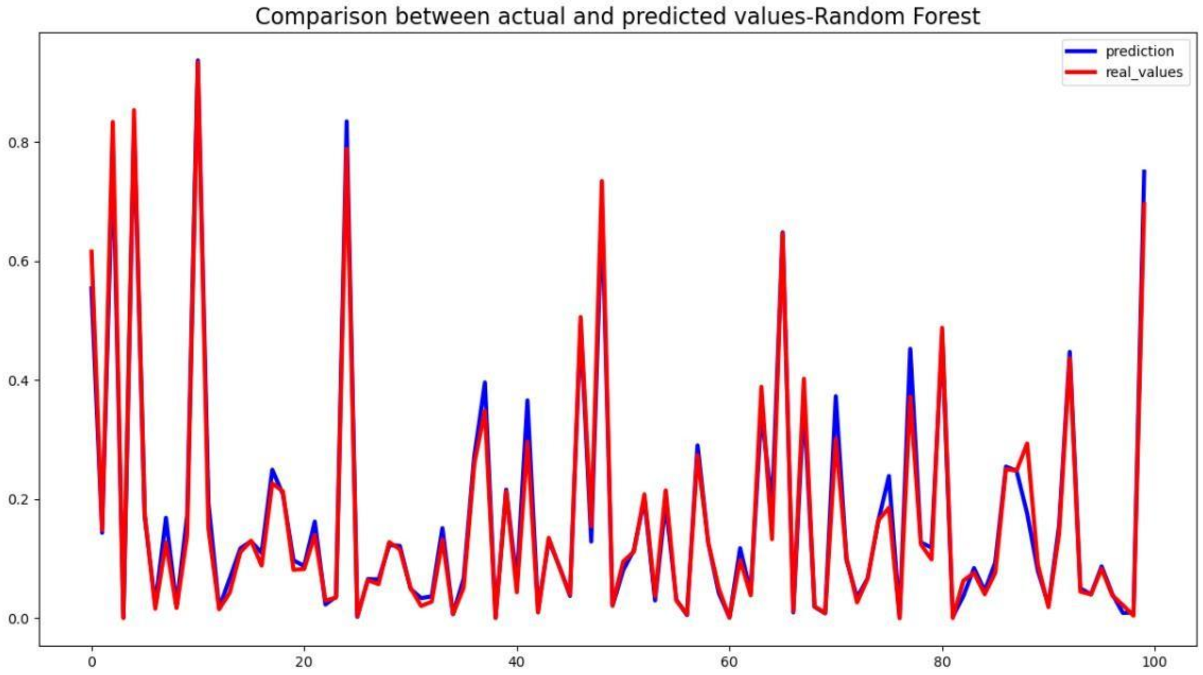


Fig 4.9 - Line Graph of Actual v/s Predicted in Random Forest

4.3 K-Nearest Neighbor

The third model employed is the KNN model, which is grounded in a type of machine learning known as instance-based learning, sometimes referred to as lazy learning or instance-based learning, involves not explicitly training the model on a dataset throughout the learning phase. Rather, during the prediction process, the model learns from the training examples and applies its knowledge to new, unobserved occurrences. Model-based techniques, such as those utilizing explicit parameterized models trained on a dataset, are frequently used to compare instance-based learning methodologies.

The k-NN method is the most popular and straightforward instance-based learning algorithm.

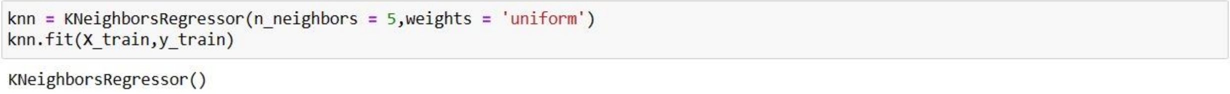
-Uses k "closest" points (nearest neighbors) for performing classification

-KNN defines neighbours in terms of distance (often Euclidean in R-space), assuming that all instances are points in some n-dimensional space.

-K is the quantity of neighbours taken into account.

-Mostly used in recommender systems and for optical character recognition (OCR).

A simple approach known as K nearest neighbours (KNN) keeps track of all examples that are accessible and defines cases that were recently discovered using a distance function. Since the 1970s, KNN has been utilized in statistical estimates and pattern recognition.



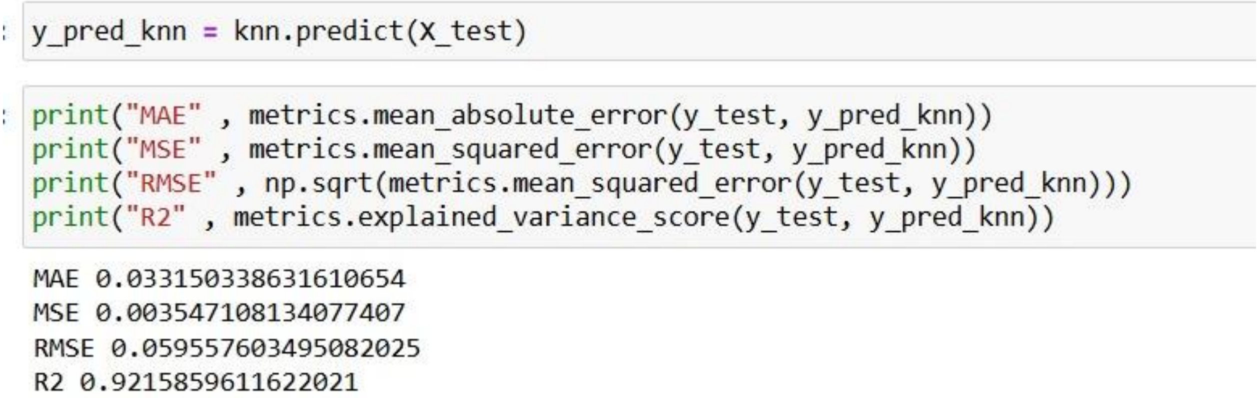
```
knn = KNeighborsRegressor(n_neighbors = 5,weights = 'uniform')
knn.fit(X_train,y_train)
```

```
KNeighborsRegressor()
```

Fig 4.10– Training K-Nearest Neighbor

We took neighbors as 5 and weights = 'uniform' in KNN mode.

Using KNN model we got **0.028 MAE**. We got **0.52 RMSE** and **0.94 R^2** score. Below is the graph of actual and predicted values using KNN model.

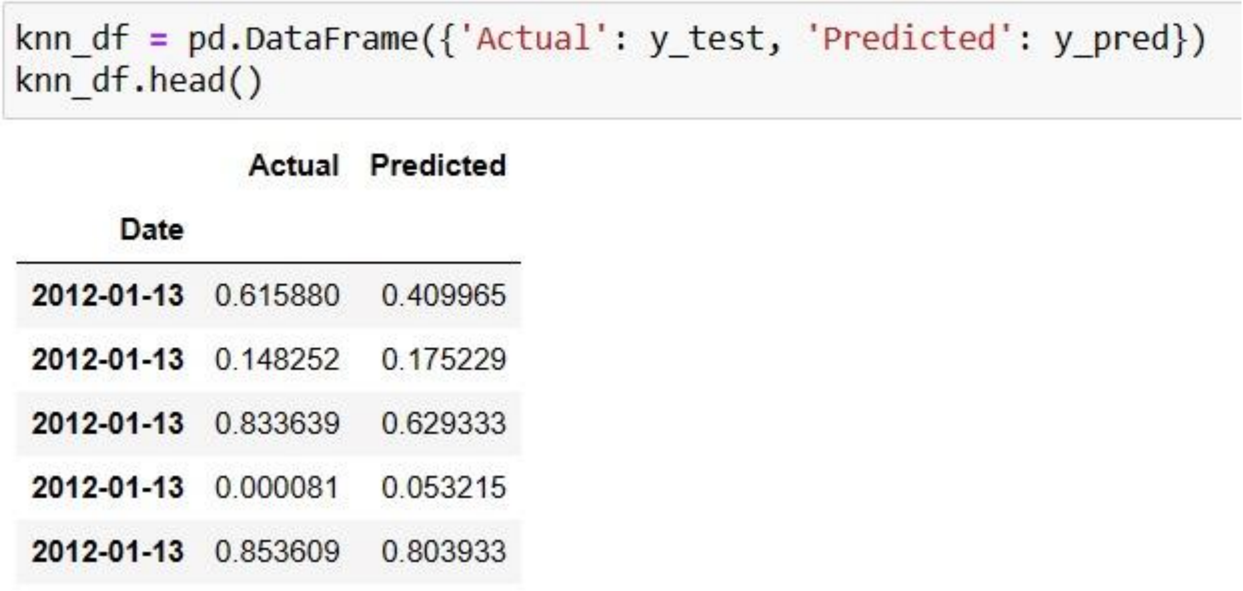


```
y_pred_knn = knn.predict(X_test)
```

```
print("MAE" , metrics.mean_absolute_error(y_test, y_pred_knn))
print("MSE" , metrics.mean_squared_error(y_test, y_pred_knn))
print("RMSE" , np.sqrt(metrics.mean_squared_error(y_test, y_pred_knn)))
print("R2" , metrics.explained_variance_score(y_test, y_pred_knn))
```

```
MAE 0.033150338631610654
MSE 0.003547108134077407
RMSE 0.059557603495082025
R2 0.9215859611622021
```

Fig 4.11– MAE, MSE, RMSE and R^2 score in KNN



```
knn_df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
knn_df.head()
```

Date	Actual	Predicted
2012-01-13	0.615880	0.409965
2012-01-13	0.148252	0.175229
2012-01-13	0.833639	0.629333
2012-01-13	0.000081	0.053215
2012-01-13	0.853609	0.803933

Fig 4.12-Actual v/s Predicted in KNN

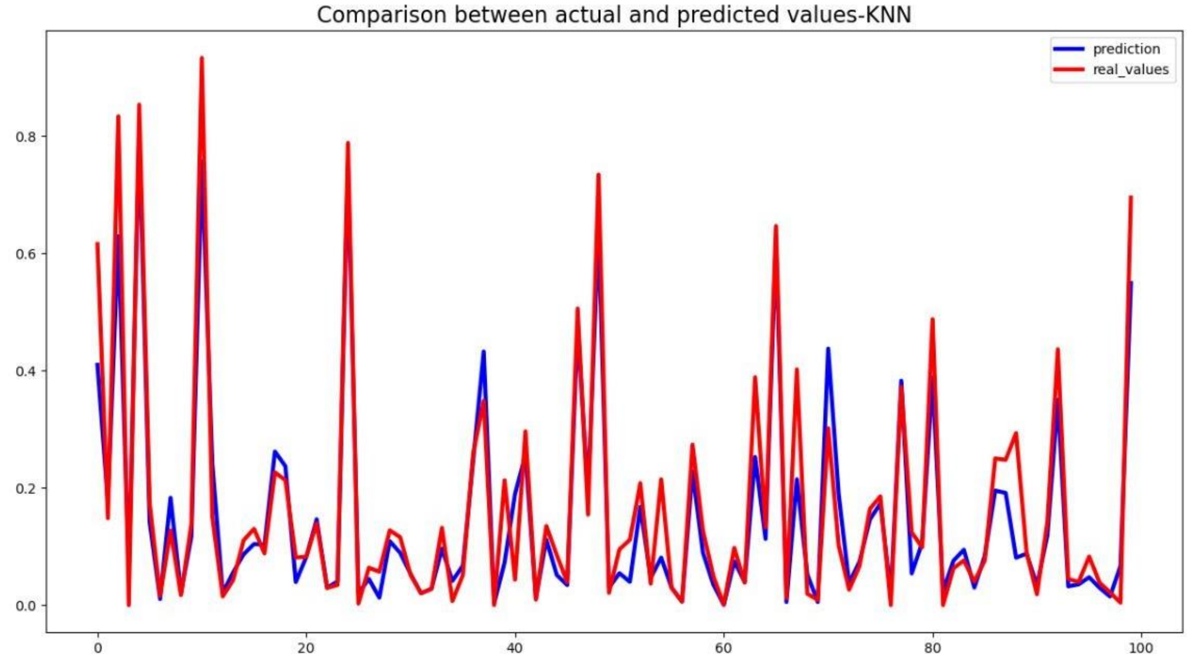


Fig 4.13 -Line Graph of Actual v/s Predicted in KNN

4.4 XGBoost Model

The last model we used is **XGBoost model**. Using this model many Kaggle users won Kaggle big competition. XGBoost model outperforms very well.

XGBoost (Extreme Gradient Boosting) is based on the principles of gradient boosting, an ensemble learning technique. The key mathematical concepts behind XGBoost include:

1. **Objective function:** XGBoost aims to maximize an objective function comprising a regularization term and a loss function. The optimization process is directed by this loss function. The optimization process is guided by the loss function, which calculates the difference between the actual and expected values. By punishing complicated models, the regularization term aids in preventing overfitting.

2. **Gradient Boosting: --** Gradient boosting is an iterative method that amalgamates the forecasts of numerous weak learners, usually decision trees, to craft a robust predictive model. To fix the mistakes in the previous model, a new weak learner is introduced in each iteration.

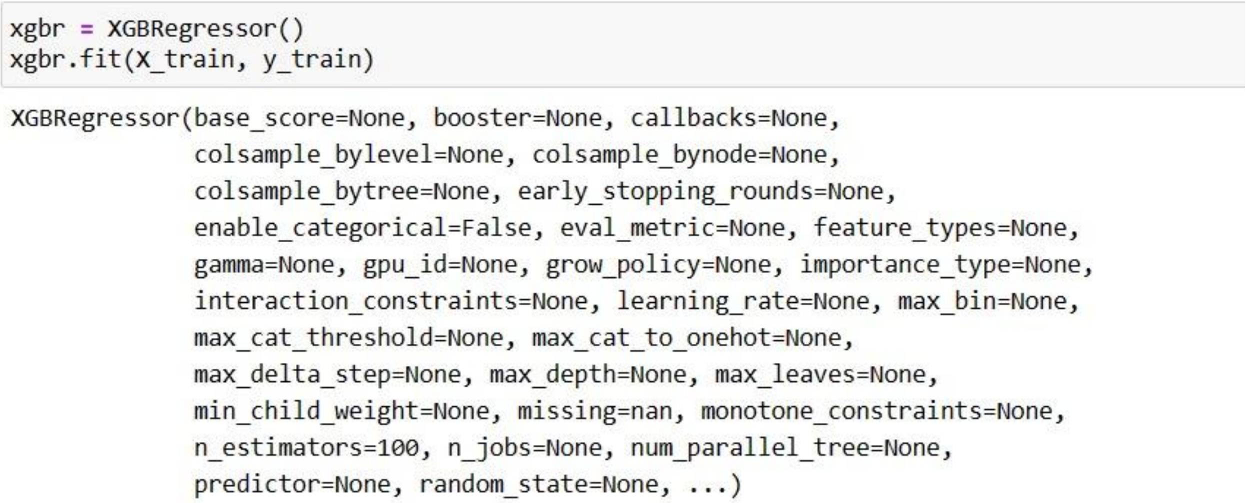
3. **Decision Trees:** Regression trees are used by XGBoost, and each tree makes a forecast predicated on a portion of the attributes. Trees are constructed one after the other, with each new tree fixing the mistakes of its predecessor.

4. **Loss Function:** The loss function is contingent upon the nature of the problem (classification or regression) and the particular objectives of the modelling assignment. For instance, logistic loss (cross- entropy) is employed for binary classification, whereas the mean squared error is frequently utilized for regression.

5. **Regularization Terms**: The objective function of XGBoost incorporates L_1 (Lasso) and L_2 (Ridge) regularization terms. By penalizing the model's big coefficient these terms lessen the chances of overfitting and maximize the model efficiency.

6. **Gradient Descent:** To minimize the objective function, XGBoost employs gradient descent optimization. The algorithm determines the gradient for each iteration updates the parameters in the direction that minimizes the objective after considering the objective function in relation to the model's parameters.

7. **Tree Pruning:** To regulate the size of the trees, XGBoost incorporates a pruning process. Pruning is the process of cutting off tree branches that don't greatly enhance the model, which keeps it from overfitting.

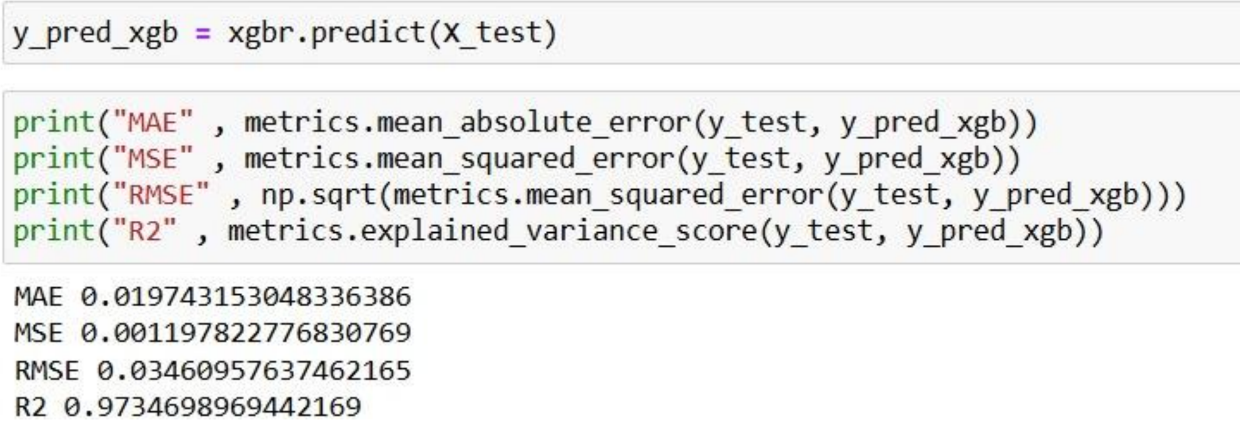


```
xgbr = XGBRegressor()
xgbr.fit(X_train, y_train)
```

```
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             n_estimators=100, n_jobs=None, num_parallel_tree=None,
             predictor=None, random_state=None, ...)
```

Fig 4.14- Training XGBoost

Using the default parameters of Xgboost model we got **0.44 RMSE** score, **0.96 R^2** score and **0.02 MAE.**

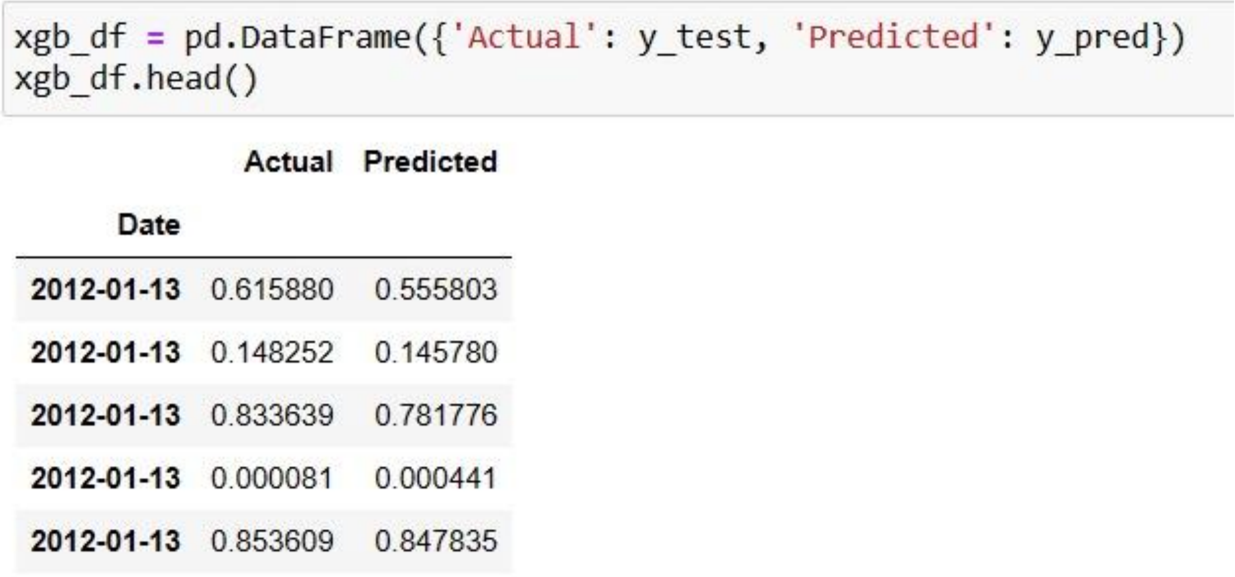


```
y_pred_xgb = xgbr.predict(X_test)
```

```
print("MAE" , metrics.mean_absolute_error(y_test, y_pred_xgb))
print("MSE" , metrics.mean_squared_error(y_test, y_pred_xgb))
print("RMSE" , np.sqrt(metrics.mean_squared_error(y_test, y_pred_xgb)))
print("R2" , metrics.explained_variance_score(y_test, y_pred_xgb))
```

```
MAE 0.019743153048336386
MSE 0.001197822776830769
RMSE 0.03460957637462165
R2 0.9734698969442169
```

Fig 4.15– MAE, MSE, RMSE and R^2 score in XGBoost



```
xgb_df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
xgb_df.head()
```

Date	Actual	Predicted
2012-01-13	0.615880	0.555803
2012-01-13	0.148252	0.145780
2012-01-13	0.833639	0.781776
2012-01-13	0.000081	0.000441
2012-01-13	0.853609	0.847835

Fig 4.16– Actual v/s Predicted in XGBoost

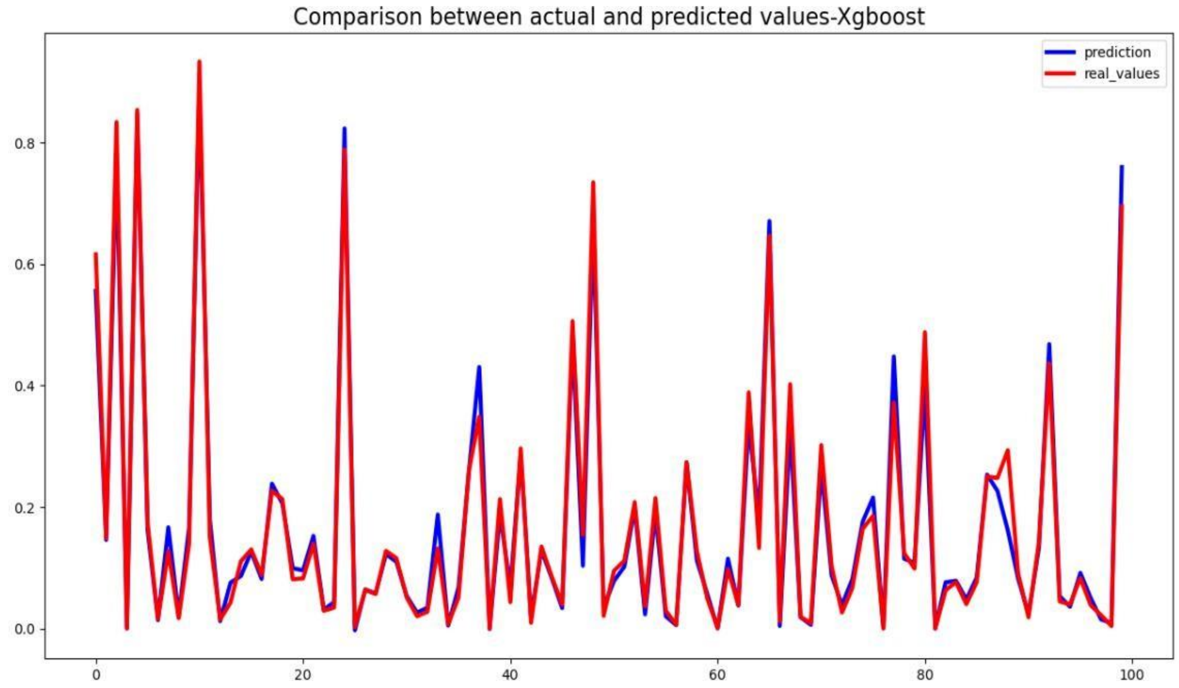


Fig 4.17- Line Graph of Actual v/s Predicted in XGBoost

4.5 Support Vector Machine

A type of learning known as Support Vector Machine (SVM) can be used for the problems involving regression or classification. But mostly SVM is used for classification. It simply divides the feature space into two groups.

The SVM method is used to depict each data item as a point in p-dimensional space and is applied to classification problems by the constraints [11].

In order to model the scenario, a Support Vector Machine creates a vector space termed a feature space with limited parameters, each of which stand for a "feature" of a specific item.

Using the Support Vector Machine (SVM), one can train a model to classify previously unknown objects into new categories.

This can be achieved by building a linear and the value of each feature is correlated with a certain place. Afterwards, we implement the classification approach by defining the hyper-plane that optimally divides the two groups.

Places an object "above" or "below" the separation plane based on features in the newly unrecognized objects (e.g. emails), leading to a categorization (e.g. spam or non-spam). That is a representation of an uncertain linear classifier as a result.

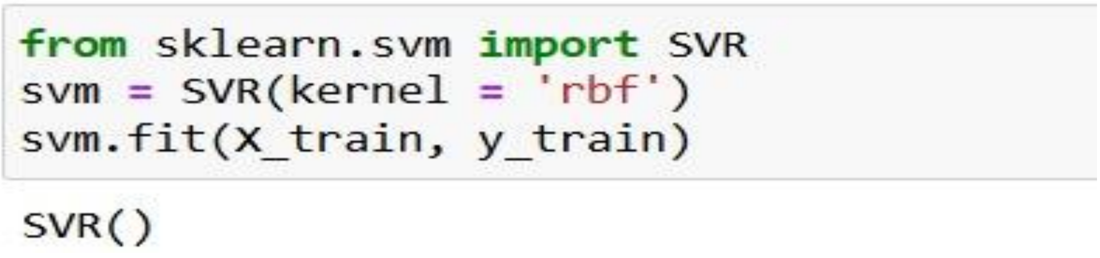
Just the coordinates of each unique observation make up a support vector.

The data points nearest to the decision surface (also called as hyperplane) are called as support vectors.

These data points pose the greatest challenge for classification and significantly influence the positioning of the decision surface. The data points represent the support vectors.

Support vectors are the points in the data set that are nearest to the hyperplane and that, would cause the dividing hyperplane to shift if eliminated.

As such, they may be thought of being the key components of the collection of information.

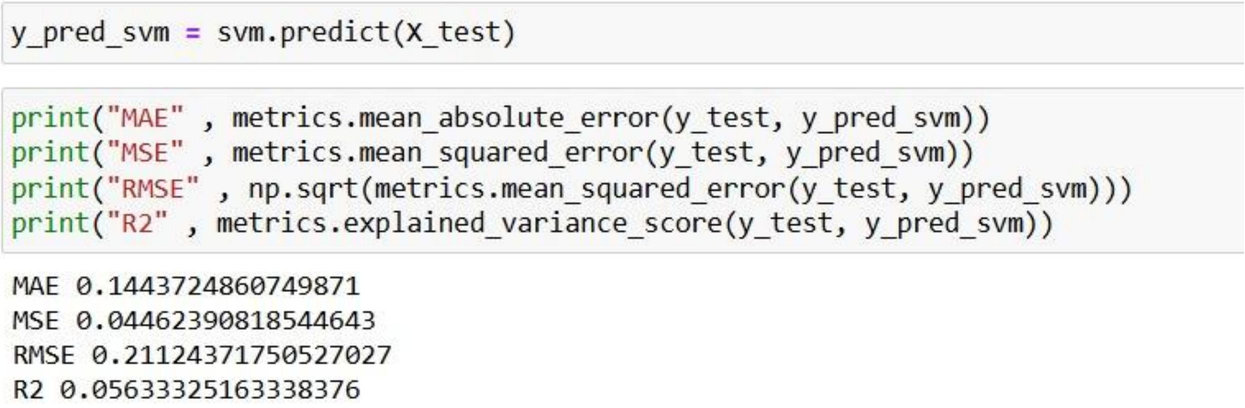


```
from sklearn.svm import SVR
svm = SVR(kernel = 'rbf')
svm.fit(X_train, y_train)
```

```
SVR()
```

Fig 4.18- Training SVM

In SVM we got **0.21 RMSE** score, **0.05 R^2** score and **0.14 MAE.**

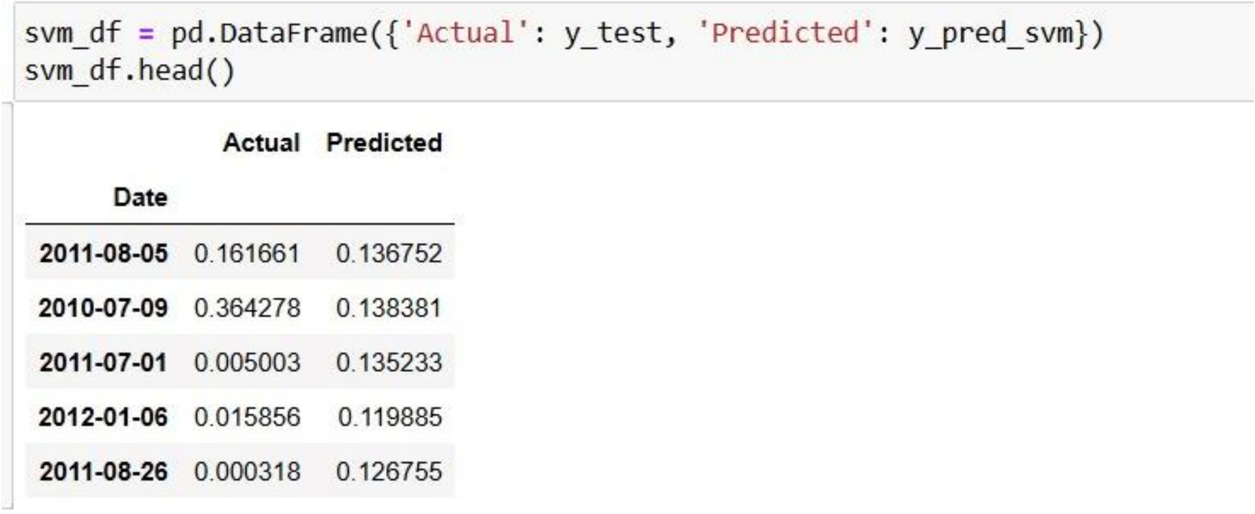


```
y_pred_svm = svm.predict(X_test)
```

```
print("MAE" , metrics.mean_absolute_error(y_test, y_pred_svm))
print("MSE" , metrics.mean_squared_error(y_test, y_pred_svm))
print("RMSE" , np.sqrt(metrics.mean_squared_error(y_test, y_pred_svm)))
print("R2" , metrics.explained_variance_score(y_test, y_pred_svm))
```

```
MAE 0.1443724860749871
MSE 0.04462390818544643
RMSE 0.21124371750527027
R2 0.05633325163338376
```

Fig 4.19 MAE, MSE, RMSE and R^2 score in SVM



```
svm_df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred_svm})
svm_df.head()
```

	Actual	Predicted
Date		
2011-08-05	0.161661	0.136752
2010-07-09	0.364278	0.138381
2011-07-01	0.005003	0.135233
2012-01-06	0.015856	0.119885
2011-08-26	0.000318	0.126755

Fig 4.20 - Actual v/s Predicted in SVM

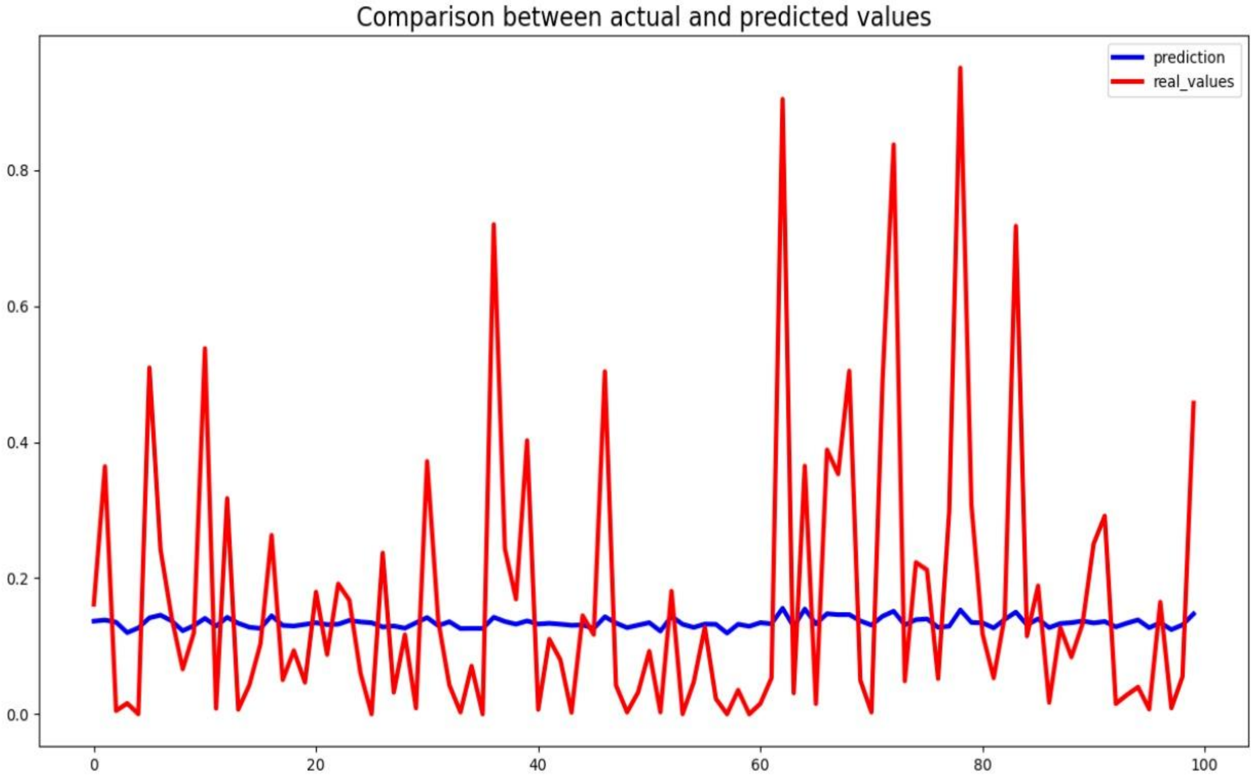


Fig 4.21 Line Graph of Actual v/s Predicted in SVM

CHAPTER 5
RESULT AND DISCUSSION

5.1 Experimental Setup

The model codes are written and executed in Jupyter Notebook and Google Colab. The system running this setup is AMD Ryzen 5 3450U with Radeon Vega Mobile Gfx 2.10 GHz with 8.00 GB RAM and system type 64-bit operating system, x64-based processor

5.2 Result Analysis

In this study of machine learning models, we evaluated the performance of XGBoost, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM), Random Forest, Linear Regression.

The primary metric used for evaluation was the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). Among the models tested, Random Forest emerged as the best performer, achieving the lowest MAE and RMSE. This indicates its superior capability in capturing complex relationships within the data while maintaining robustness against overfitting.

These results suggest that Random Forest performs well with this dataset, likely because of its capacity to handle non-linear relationships and feature interactions. The study highlights the importance of model selection based on specific performance metrics and dataset characteristics to achieve optimal predictive accuracy.

Table 5.1-Performance of Machine Learning Models

Approach	MAE	MSE	R2	RMSE
Linear Regression	0.02933	0.00288	0.93634	0.05375
Random Forest	**0.02058**	**0.00143**	**0.96783**	**0.03794**
KNN	0.02861	0.00270	0.94073	0.05203
XGBoost	0.02240	0.00160	0.96410	0.04004
SVM	0.14437	0.04462	0.05633	0.21124

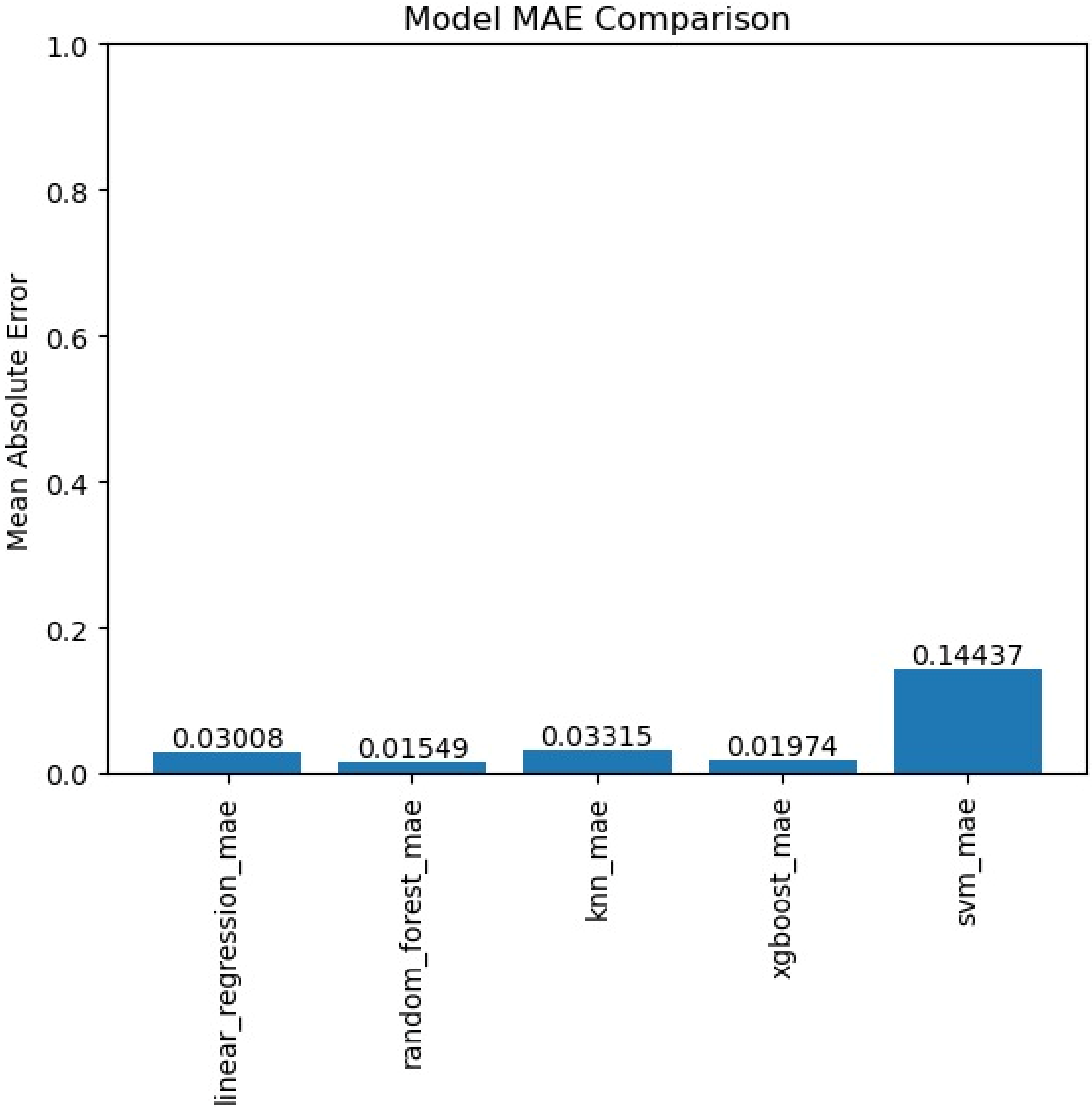


Fig 5.1 – MAE comparison of all the 5 models

As We can see in Table 5.1, Random Forest has least MAE and RMSE than other machine learning models.

XGBoost also performed well, achieving the second-best results across all metrics, followed closely by KNN and Linear Regression, which showed competitive but slightly lower performance. SVM significantly underperformed compared to the other models, with much higher error metrics and a considerably lower R^2 value, indicating poor fit and predictive capability for this dataset.

These results underscore the efficacy of ensemble methods such as Random Forest and XGBoost in managing intricate data patterns, rendering them preferred options for attaining high predictive accuracy. Future work can further refine these models and explore their integration with other advanced techniques to enhance performance and applicability.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

In conclusion, the comparative study of machine-learning models highlights the strengths and weaknesses of various algorithms in different contexts. Models such as Linear Regression, XGBoost, K-nearest neighbor, Support Vector Machines, and Random Forest exhibit unique performance characteristics, suggesting that model selection should be tailored to specific data characteristics and problem requirements. The analysis underscores the importance of using appropriate evaluation metrics and cross-validation techniques to ensure robust performance assessment.

For future research, expanding the dataset diversity and incorporating more complex and hybrid models could yield deeper insights. Exploring advancements in deep learning and time series analysis models could further enhance predictive accuracy and model robustness. Additionally, integrating interpretability techniques will be crucial for the practical deployment of these models, ensuring that their decisions can be understood and trusted by users across various domains. Continued innovation in this field promises to significantly advance the efficacy of machine learning applications.

REFRENCES

[1] Bohdan M. Pavlyshenko, 18 January 2019, Machine-Learning Models for Sales Time Series Forecasting

[2] Singh Manpreet, Bhawick Ghutla, Reuben Lilo Jnr, Aesaan FS Mohammed, and Mahmood A. Rashid. "Walmart's Sales Data Analysis-A Big Data Analytics Perspective." In 2017 4th Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE), pp. 114-119. IEEE, 2017.

[3] J Saran Prakash, Dr. Swati sah, 2023 JETIR March 2023, Volume 10, Issue 3, Big Data Analytics and Machine Learning for Sales Prediction: A Comparative Study of Traditional and Modern Approaches

[4] Asmita Manna, Kavita Kolpe, Aniket Mhalungekr, Sainath Pattewar, Pushpak Kaloge, Ruturaj Patil, 2023, Comparative Study of Various Machine Learning Algorithms for Pharmaceutical Drug Sales Prediction

[5] Rao Faizan Ali; Amgad Muneer; Ahmed Almaghthawi; Amal Alghamdi; Suliman Mohamed Fati; Ebrahim Abdulwasea; Abdullah Ghaleb, IAES International Journal of Artificial Intelligence (IJ-AI) Vol. 12, No. 2, June 2023, pp. 874~883 ISSN: 2252-8938, DOI: 10.11591/ijai.v12.i2.pp874-883, BMSP-ML: big mart sales prediction using different machine learning techniques

[6] Sathyanarayana S, Apeksha C, Chethana S, Chinmayee H C, Abhishree G L International Journal of Advanced Research in Computer and Communication Engineering ISO 3297:2007 Certified|Impact Factor 8.102|Vol. 12, Issue 4, April 2023, BIG MART SALES PREDICTION USING MACHINE LEARNING

[7] Gopal Behera and Neeta Nain, September 2019, A Comparative Study of Big Mart Sales Prediction

[8] Sushila Ratre & Jyotsna Jayaraj, 01 January 2023, Sales Prediction Using ARIMA, Facebook’s Prophet and XGBoost Model of Machine Learning

[9] Younes Ledmaouia∗, Adila El Maghraouib, Mohamed El Aroussia, Rachid Saadanea, Ahmed Chebakb, Abdellah Chehric, 22 July 2023, Forecasting solar energy production: A comparative study of machine learning algorithms 13. Jiaming Chen, 27 September 2023, Comparative Analysis of Machine Learning Models for Lung Cancer Prediction

[10] Jiaming Chen, 27 September 2023, Comparative Analysis of Machine Learning Models for Lung Cancer Prediction

[11] dspace.dtu.ac.in:8080

Similarity Report

PAPER NAME

Dissertation Final (2) (1).pdf

AUTHOR

. .

WORD COUNT

5986 Words

CHARACTER COUNT

33057 Characters

PAGE COUNT

42 Pages

FILE SIZE

1.5MB

SUBMISSION DATE

Jun 1, 2024 7:55 PM GMT+5:30

REPORT DATE

Jun 1, 2024 7:56 PM GMT+5:30

● 9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 4% Internet database
- Crossref database
- 8% Submitted Works database
- 2% Publications database
- Crossref Posted Content database

● Excluded from Similarity Report

- Bibliographic material
- Cited material
- Quoted material
- Small Matches (Less then 10 words)

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CERTIFICATE OF THESIS SUBMISSION FOR EVALUATION

1. Name: Diksha Bhati and Himani
2. Roll No.: 2K22/MSCMAT/09 and 2K22/MSCMAT/17
3. Thesis title: "Analysis Performance: A Comparative Study of Machine Learning Models".
4. Degree for which the thesis is submitted: M.Sc. Mathematics
5. Faculty of the University to which the thesis is submitted: Anjana Gupta
6. Thesis Preparation Guide was referred to for preparing the thesis. YES □ NO □
7. Specifications regarding thesis format have been closely followed. YES □ NO □
8. The contents of the thesis have been organized based on the guidelines. YES □ NO □
9. The thesis has been prepared without resorting to plagiarism. YES □ NO □
10. All sources used have been cited appropriately. YES □ NO □
11. The thesis has not been submitted elsewhere for a degree. YES □ NO □
12. All the correction has been incorporated. YES □ NO □
13. Submitted 2 hard bound copies plus one CD. YES □ NO □

(Signature of Candidate)
Name(s): Diksha Bhati and Himani
Roll No.: 2K22/MSCMAT/09 and 2K22/MSCMAT/17

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CERTIFICATE OF FINAL THESIS SUBMISSION

1. Name: Diksha Bhati and Himani
2. Roll No.: 2K22/MSCMAT/09 and 2K22/MSCMAT/17
3. Thesis title: "Analysis Performance: A Comparative Study of Machine Learning Models".
4. Degree for which the thesis is submitted: M.Sc. Mathematics
5. Faculty of the University to which the thesis is submitted: Anjana Gupta
6. Thesis Preparation Guide was referred to for preparing the thesis. YES □ NO □
7. Specifications regarding thesis format have been closely followed. YES □ NO □
8. The contents of the thesis have been organized based on the guidelines. YES □ NO □
9. The thesis has been prepared without resorting to plagiarism. YES □ NO □
10. All sources used have been cited appropriately. YES □ NO □
11. The thesis has not been submitted elsewhere for a degree. YES □ NO □
12. All the correction has been incorporated. YES □ NO □
13. Submitted 2 hard bound copies plus one CD. YES □ NO □

Signature of Supervisor
Prof. Anjana Gupta

Signature of Candidates
Name(s)- Diksha Bhati and Himani
Roll no.- 2K22/MSCMAT/09 and 2K22/MSCMAT/17