Group: Xiaoqing Xia, Digvijay Yadav

Walmart Store Weekly Sales Predicting

Abstract

The Walmart Weekly Sales Predicting Project aims to apply data analysis techniques to understand and predict the trends in weekly sales, providing valuable insights into customer needs and preferences this paper focuses on predicting the weekly sales prediction of Walmart Retail Stores using ARIMA, Linear Regression, Isolation Forests, Random Forests, Xtreme Gradient Boosting algorithms and an ensemble between Random Forest regressor and Xtreme Gradient Boosting regressor called Voting Regressor, using lag variables of 3 weeks to predict weekly sales of all the stores and departments. Models used were evaluated using Mean Absolute Error (MAE), and Coefficient of determination scores. Factors like Temperature and CPI were considered for identifying potential associations between these variables and the outcome of weekly sales. Adding lag variables to the dataset enhanced the performance of the models used, with the Random Forest Model achieving the lowest MAE score of 2514, and a Coefficient of determination score of 0.94. We were also able to conclude that the initial assumption that Temperatures and CPI, influenced the outcome of Weekly sales were inaccurate and that the Department, Size of the stores influenced the outcome of Weekly sales based on the analysis from the Machine Learning models.

1. Introduction

1.1 Project Background

With advances in technology, companies have access to a wealth of data that holds great potential for extracting valuable insights. Data analysis has become instrumental in guiding businesses toward informed decision-making and devising successful marketing strategies. In our daily lives, retail stores play a significant role as they provide essential goods such as vegetables, furniture, electronics, and much more. The presence and importance of retail stores in our lives have sparked a curiosity to delve deeper into analyzing the trends within this sector. As a result, this project focuses on examining the weekly sales trends of Walmart, one of the largest and most influential retail chains globally.

The well-known American multinational retailer Walmart is a major participant in the retail sector, running a variety of supermarkets, outlet shops, and grocery stores throughout the country (*Walmart*). Both merchants and customers must understand the dynamics of the weekly sales. From the perspective of merchants, they can analyze the sales trends of customer preferences, market demands, and the effectiveness of marketing strategy. Specifically, Walmart can leverage the knowledge of weekly sales trends to efficiently manage inventory, optimize product organization, establish competitive pricing, and develop effective promotion strategies.

The data in this project is about the sales of Walmart stores from the Kaggle website and contains 420209 data in total with 45 stores (Cukierski). By leveraging historical sales data, of around 2 years from 2010 to 2012 along with relevant external factors, the project aims to provide valuable insights to improve the efficiency of the supply chain and minimize the overstock and stockout using the ARIMA model and machine learning methods. The project also

aims to figure out the importance of features: Temperature, Fuel_Price, Markdown, CPT, Unemployment, Store type, and Store Size (Jeswani 12). Our research utilizes three distinct datasets. The first dataset, known as the training dataset, encompasses a comprehensive collection of historical data spanning from 2010 to 2012. This invaluable resource provides us with detailed insights into the weekly sales information, serving as a foundation for our investigation. The second dataset, aptly named "Store," contains fundamental information about the various stores under examination. Within this dataset, we can explore vital details such as the store type and size. These factors are essential in discerning the unique characteristics and attributes associated with each store, enabling us to draw meaningful conclusions within our research. Lastly, we have the Feature dataset, which encompasses a diverse range of external factors that exert influence on our analysis. These factors encompass elements such as Temperature, IsHoliday, CPI (Consumer Price Index), Fuel Price, and more. By incorporating this dataset into our study, we aim to delve deeper into the relationship between these external factors and the weekly sales figures, unraveling the intricate interplay between them.

Jeswani, in addition to other contributions, authored the report on the forecast of Walmart sales. In the report, the author first drew the data visualization between IsHoliday and Weekly Sales (Jeswani 33). During his analysis, he made a noteworthy observation indicating that the average sales during holiday periods significantly surpass the average sales during nonholiday periods, even though the total number of days designated as holidays is substantially fewer than the count of nonholiday days. He proposed the introduction of a weighted mean absolute error (WMAE) metric that specifically incorporates the influential factor of holidays to accurately assess the absolute error of various prediction models (Jeswani 38). Through his research, Jeswani made a significant discovery by utilizing the linear regression model as the baseline for

his analysis. He employed the weighted mean absolute error calculation to evaluate the performance of this baseline model. Jeswani further investigated and applied additional models, aiming to identify alternative models that yielded the lowest WMAE scores (Jeswani 38). In our research, we aim to explore the correlation between the IsHoliday variable and Weekly Sales, to determine the most suitable metric for assessing the performance of different models. By analyzing the relationship between these two factors, we seek to uncover valuable insights regarding the impact of holidays on sales and identify the metric that will effectively evaluate and compare the performance of various models in our study.

Accurate projections of future retail sales can improve the efficiency of operations within the retail sector and its supply chains. As a result, many local and foreign experts have been interested in sales volume forecasting using models. The following research findings are multiple linked findings. Zhang Chu used time series methods and regression methods for projecting total retail sales (Yi 87). The findings demonstrate that, when applied to datasets with significant seasonal fluctuation, neural network model performance outperforms both alternative regression approaches and time series methods. To project electricity sales for power companies, Zhao designed a visual clustering technique, a specialized regression methodology, and a time series method (Yi 87). By using clustering and classification algorithms, Thomassey, and Fiordaliso forecasted textile sales (Yi 87). Furthermore, Gupta employed a range of machine learning methods and implemented hyperparameter tuning techniques to identify the optimal model with the lowest weighted mean absolute error (WMAE) score (Gupta).

2. Methods

2.1 Data Cleaning and Exploration

There are three datasets for this project: train, features, and stores. In the training dataset are Store, Dept, Date, Weekly Sales, and IsHoliday. Figure 2.1.1 shows the information in the training dataset and the IsHoliday variable identifies whether the specific data is a holiday or not.

```
In [8]: train.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 421570 entries, 0 to 421569
Data columns (total 5 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   Store         421570 non-null  int64
 1   Dept          421570 non-null  int64
 2   Date          421570 non-null  object
 3   Weekly_Sales  421570 non-null  float64
 4   IsHoliday     421570 non-null  bool
dtypes: bool(1), float64(1), int64(2), object(1)
memory usage: 13.3+ MB
```

Figure 2.1.1 Information in train dataset

In the stores' dataset, they are Store, Type, and Size. This dataset contains the store's information including the Size and the Type of the Store. In the features dataset, they are Store, Date, Temperature, Fuel_Price, MarkDown1~MarkDown5, CPI, Unemployment, and IsHoliday. These columns represent the external factors that will affect the retail sales of the Walmart Store. To analyze the factors that have the most impact on weekly sales, we performed data integration using the Stores dataset and the training dataset, both of which include the 'Store' variable. By performing an inner join on the Store variable, we combined the two datasets and added two additional variables to the training dataset. Next, we further enriched the training dataset by performing an inner join with the features dataset, using the Store, IsHoliday, and Date variables. This allowed us to combine the information from the features dataset with the training dataset. The resulting train dataset now contains all the relevant information necessary for analyzing the

factors that influence weekly sales the most shown in Figure 2.1.2. We can now proceed with the analysis to determine the importance of these factors in predicting weekly sales (Jeswani 12).

```
In [13]: train.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 421570 entries, 0 to 421569
Data columns (total 16 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   Store          421570 non-null  int64
 1   Dept           421570 non-null  int64
 2   Date           421570 non-null  object
 3   Weekly_Sales   421570 non-null  float64
 4   IsHoliday      421570 non-null  bool
 5   Temperature    421570 non-null  float64
 6   Fuel_Price     421570 non-null  float64
 7   MarkDown1      421570 non-null  float64
 8   MarkDown2      421570 non-null  float64
 9   MarkDown3      421570 non-null  float64
 10  MarkDown4      421570 non-null  float64
 11  MarkDown5      421570 non-null  float64
 12  CPI            421570 non-null  float64
 13  Unemployment   421570 non-null  float64
 14  Type           421570 non-null  object
 15  Size           421570 non-null  int64
dtypes: bool(1), float64(10), int64(3), object(2)
memory usage: 51.9+ MB
```

Figure 2.1.2 Information of Combined Dataset

As is the cardinal step with any Machine Learning approaches, the first step is to understand the dataset and check for the distribution of the dataset if there are any missing values or outliers that are present in the dataset. We proceeded with checking the distribution of the dataset. Most of the features had skewed distributions indicating that overall the dataset was not normally distributed, while Temperature, Fuel Price, Unemployment, and Store had distributions closer to a normal distribution, the rest did not. Figure 1.2.3 below shows the distribution of individual columns.

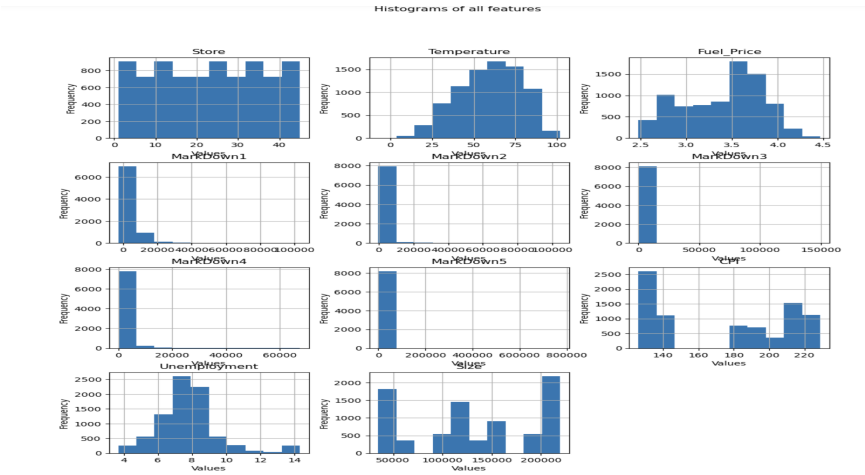

Figure 2.1.3 Histogram by Columns

We then proceeded to identify the missing values in the dataset, if any existed, as these missing values can severely affect the modeling performance of the algorithms we use. Based on the analysis we found out that columns MarkDown 1 ~ MarkDown 5 had the majority of the missing values in the dataset. Other columns that had missing values were CPI and Unemployment. Any columns with missing values of more than 90% are usually discarded as nothing meaningful can be derived from them even after imputing values. Figure 1.2.4 below shows the missing value proportions per column. Here the columns had around 50-60% of missing values, so we proceeded with imputing them. Given the Markdown variables represent the promotive activities throughout the year, the missing values should be considered as 0 when there are many missing values. For Markdown columns, we imputed the missing values by replacing NaN with zeros. Filling the Markdown columns with zeros was done to ensure that there is no promotional activity. Since we were focusing on predicting the weekly sales from different features, preserving the data structure was important, and hence we preferred filling them with zeros. Columns CPI and Unemployment were alternatively filled with median values to prevent any biases in the dataset based on the distribution of these columns.

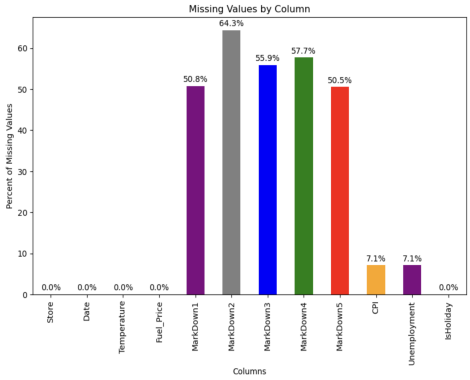

Figure 2.1.4 Missing Percentage by Column

After we imputation missing values from the dataset, we summarized the dataset to understand the spread of values in the train dataset, here we can see that all of the columns have 421570 values. However, if the minimum value of Weekly Sales is smaller than 0, then we remove the negative Weekly Sales from the dataset.

2.2 Exploratory Data Analysis

After processing the dataset, we explore the fundamental relationship between various features and Walmart's weekly sales. One of our primary interests was determining whether there is a significant difference in sales between holidays and non-holidays. To investigate this, we created a plot (Figure 2.2.1) that illustrates the association between weekly sales and the IsHoliday feature. We observed that the average weekly sales during holidays were slightly higher compared to non-holiday periods from this plot. Specifically, the average sales during holidays were approximately 1.07 times higher than during non-holiday periods. This represents that there is no significant relationship between IsHoliday and Weekly Sales.

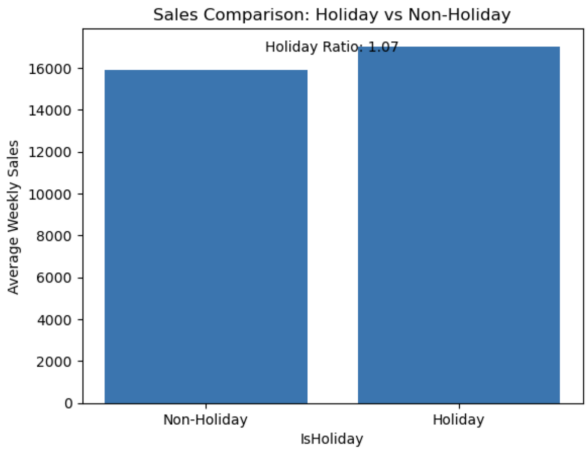

Figure 2.2.1 IsHoliday vs Weekly Sales

There are three types of stores: ‘A’, ‘B’ and ‘C’. We are considering the potential correlation between Store Type and Size of the Store. By grouping the store based on their store type and calculating the cumulative size of each group, the plot shown in Figure 2.2.2 indicates the clear relationship between Store Size and Store Type. Store A exhibits the largest Store Size among all the types.

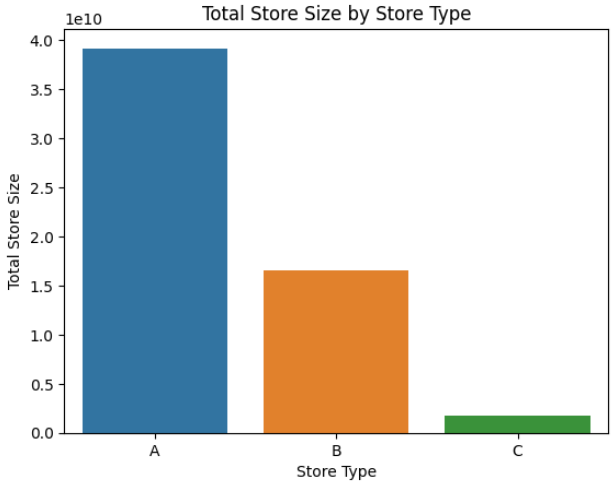

Figure 2.2.2 Store Size by Store Type

To examine the relationship between Store Type and Weekly Sales, we plot the bar plot that illustrates the Average Sales for each Store Type (Figure 2.2.3). After observing the sales trend, there is a decreasing trend of sales in Store A, Store B, and Store C. This represents that Store Type impacts the Average Sales to an extent. Based on the previous analysis of store size and store type, it can be illustrated that there exists a direct relationship between Store Size and Average Sales.

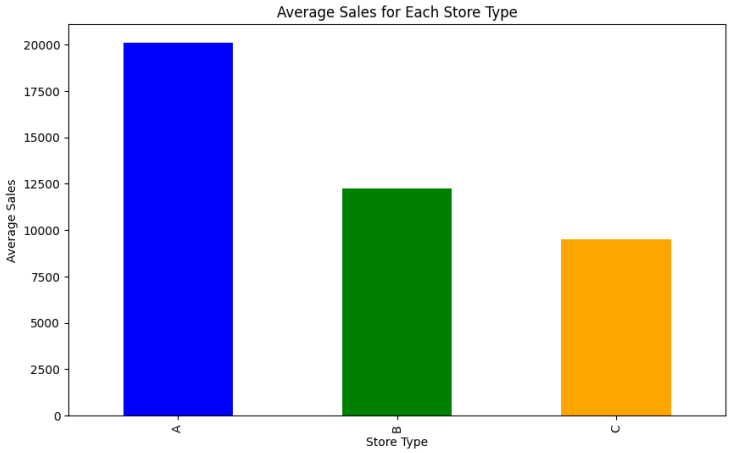

Figure 2.2.3 Store Type and Weekly Sales

The Heatmap shows the correlation matrix of each column in the dataset. Heatmap is useful when dealing with a dataset that contains many features, which will provide a comprehensive overview of the entire dataset. Figure 2.2.4 shown below identifies the strength and the direction of the relationship between different columns. Especially, the color distinct from the intensity of the correlation for each relationship. From this plot, we can observe size and department have a relationship with Weekly Sales. The larger store size and certain kinds of departments may have a stronger impact on sales. The remaining features display small correlations with Weekly Sales.

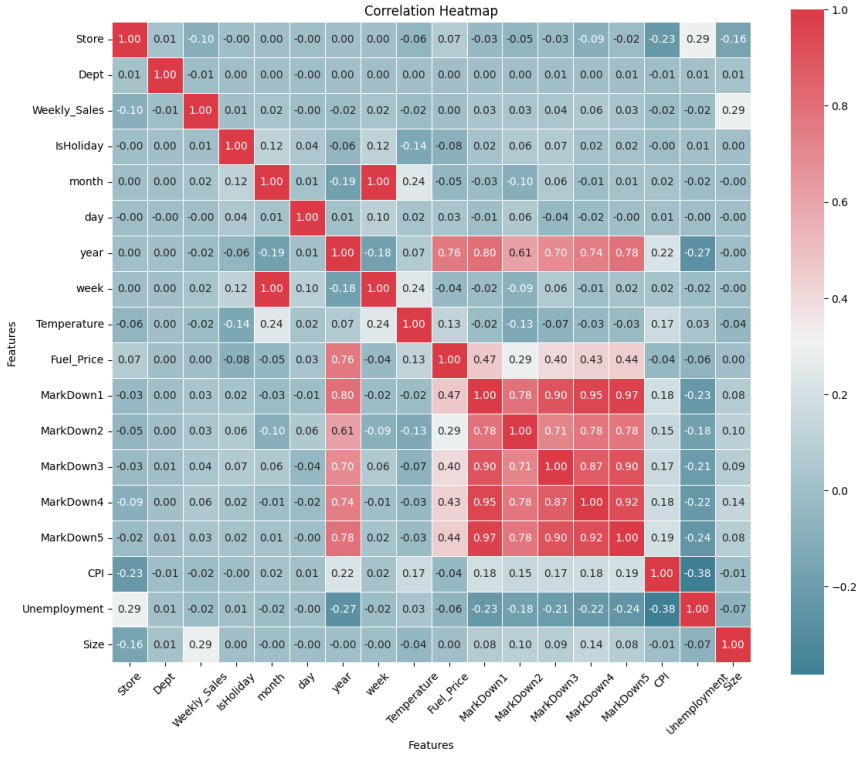

Figure 2.2.4 Correlation HeatMap

2.3 ARIMA Modeling

The ARIMA model is a widely used time series forecasting method specifically designed for datasets that exhibit seasonal patterns. It enables us to make predictions about future outcomes based on the historical patterns found within the data. Walmart is suitable to use the ARIMA model to predict the trend of Weekly Sales. By applying the ARIMA model, we can capture the pattern and trend of the sales series. Overall the ARIMA model can provide valuable insights into Walmart's weekly sales pattern.

The seasonal plot of Walmart's weekly sales is created by grouping the data by Year and Week and calculating the mean of the weekly sales. From this plot, an interesting pattern emerges: there is a local maximum in sales every three weeks. Additionally, we observe that the weekly sales before 45 weeks are lower compared to those after 45 weeks. Especially, the weeks

corresponding to the Thanksgiving and Christmas holidays fall within the range of 45 to 52 weeks. The Seasonal plot is shown below in Figure 2.3.1. Upon the seasonal plot, there shows a discernable trend persisting consistently in each year.

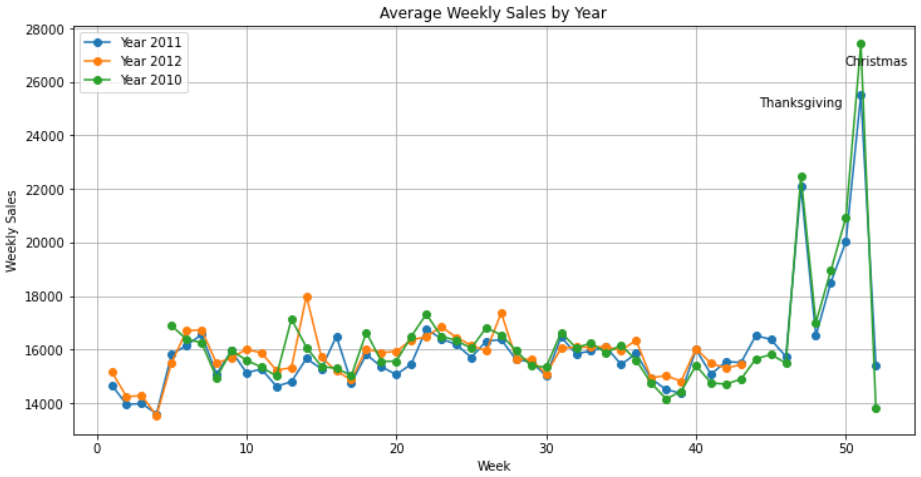

Figure 2.3.1 Seasonal Plot

The seasonal plot helps us understand the seasonal pattern of the Weekly Sales from 2010 to 2012. It's crucial to apply the ARIMA model to predict the Walmart Store Weekly Sales across time considering the presence of a distinct seasonal pattern within the Walmart Store Weekly Sales data. To identify the representative store that captures the overall characteristics of the dataset, We begin by plotting the total weekly sales of each Store and sorting them in descending order based on their respective totals, which is shown in Figure 2.3.2. We pick the store which has the highest total weekly sales which can represent the whole dataset. To determine a suitable department for implementing the ARIMA model, we group the data by department within Store 20. By calculating the cumulative weekly sales for each department, we can evaluate their respective performance. Upon analysis, we identify department 92 as the most appropriate candidate for utilization in the ARIMA model. Overall, we will use department 92 in Store 20 for implementing the ARIMA model.

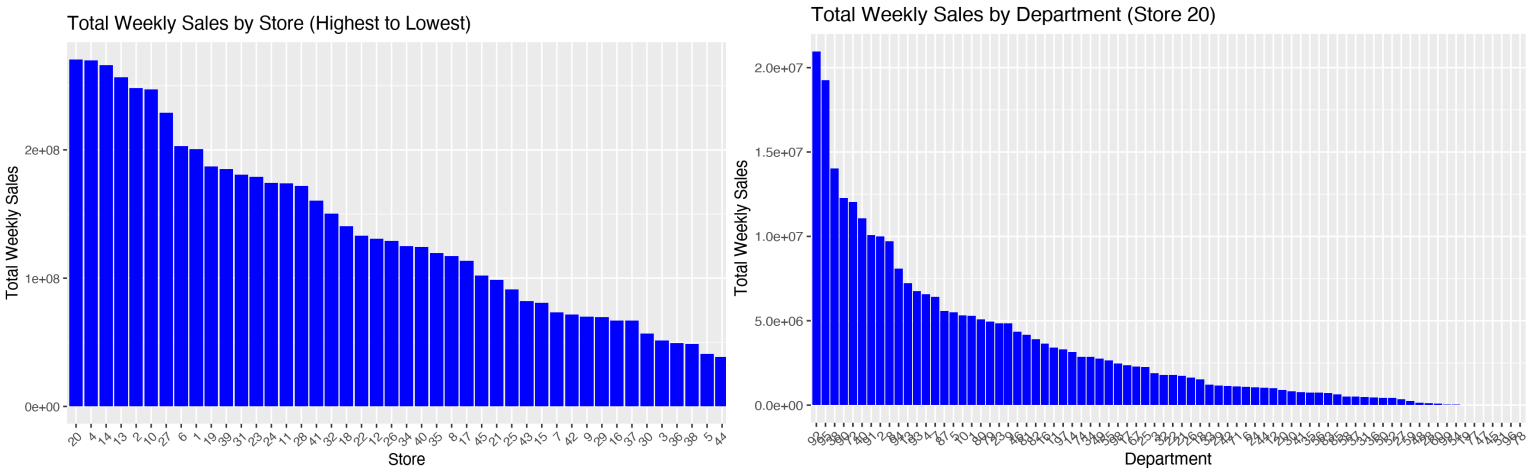

Figure 2.3.2 Total Weekly Sales by Store and Department

To assess the stationarity of the dataset, we focus on the subset consisting of the 92nd department within the 20th store. By plotting the weekly sales over time, we aim to examine the data's characteristics. Figure 2.3.3 reveals that the weekly sales exhibit a range between 0 and 225,000, displaying a noticeable increasing trend from the beginning to the end of the period. In an attempt to standardize the variance within the dataset, a log transformation is applied. Figure 2.3.3 illustrates the results, showcasing the transformed dataset. However, despite the transformation, the increasing trend remains evident in the data, indicating that the log transformation alone does not eliminate the underlying trend. By taking the first-order difference of the log-transformed data, we aim to eliminate the trend component. This differentiation process successfully removes the trend, leading to a stationary dataset that exhibits no discernible trend over time in Figure 2.3.3. In addition, the Augmented Dickey-Fuller Test is utilized to examine the stationarity of the transformed dataset. This statistical test helps determine whether the dataset displays stationary characteristics or not. In our analysis, the test result is 0.01 which is smaller than 0.05. It indicates that the transformed dataset indeed exhibits stationarity, further validating the effectiveness of the differentiation process in removing any remaining trends or non-stationary patterns.

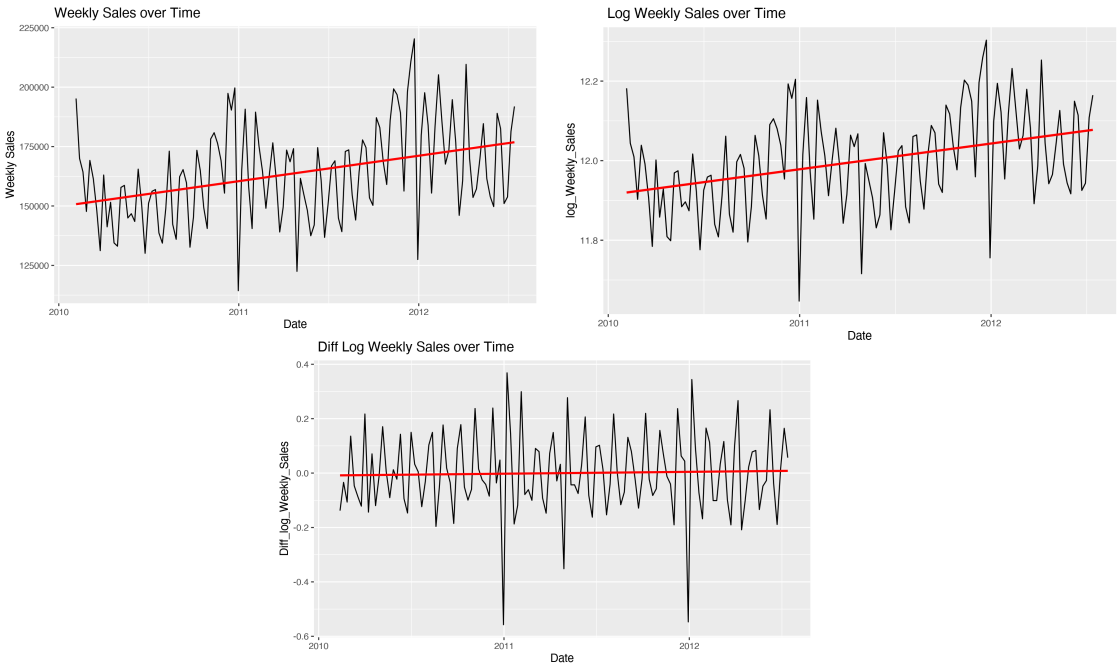

Figure 2.3.3 Original vs Log Transformation

2.4 Machine Learning-Based Approaches

The combined dataset used for further analysis consisted of 16 features with 421570. After removing missing values, and checking for duplicates in the dataset, we then proceeded with performing Machine learning modeling of the dataset. To check whether the lagged variables had any significant association with the outcome of weekly sales, we created lagged variables of Weekly Sales, of 3 weeks of data. We used similar lag variables as suggested by the ARIMA model, as it delivered better results. Using the conventional approach of data splitting, we split the dataset into training and testing. 20% of the dataset was allocated to testing. We worked on developing algorithms like Linear Regression, Random Forest Modeling, Isolation Forest Modeling, XGBoost Modeling, and Voting Regressor.

1. Simple Linear Regression

The linear Regression model is the type of machine learning algorithm that is used to estimate and predict the relationship between predictors or independent variables and outcome or target variables. It is typically used in a regression analysis where the outcome variables are continuous. Where conventional machine learning algorithms fail in being interpretable, linear regression models ofter are easier to interpret, and it assumes that all the features are independent of each other and there is a linear relationship between the independent and dependent variables. (Yi 87).

2. Random Forest Algorithm

It is one of the most popular algorithms used for tasks ranging from classification to regression. Its robustness to outliers, and ability to capture nonlinear relationships in the dataset, make it the most reliable algorithm for analysis. Since our dataset had a nonparametric distribution and as was observed from the performance of the linear regression algorithm, we decided to use the Random Forest Algorithm for predicting the Weekly sales of the Walmart data. Previous works involving Random Forests in predicting sales were in predicting house prices (Ajala et al., 808).

The algorithm is an ensemble of Decision Trees, that are trained on multiple random subsets of the dataset with replacement. The output is taken from the majority of the decision trees by averaging their votes for regression approaches. It is a type of Bootstrapping algorithm. We trained the random forest model on a similar dataset with lagged variables of 3 weeks. We also worked on exploring and tuning different hyperparameters in the model and ultimately decided to keep the estimators at 300, and we also included the random_state parameter to provide reproducible results. The package used for random forest model training was Scikit

Learn. We additionally created a feature map representing what features were important to the random forest model in making decisions.

3. XGBoost Regressor Model

Extreme Gradient Boosting (XGBoost) is a type of gradient boosting algorithm that uses decision trees as base learning algorithms and builds upon them sequentially, it can identify missing values in the dataset, and can effectively mitigate the overfitting through regularization parameters, unlike Random Forest algorithms (Chen 1).

Based on the gradient boosting principle, XGBoost combines many weak predictive models, often decision trees, to produce a powerful predictive model. The algorithm produces an ensemble of decision trees in a sequential fashion, with each new tree correcting the errors of the preceding ones. The model's overall forecast accuracy is boosted by this iterative procedure. The fact that XGBoost can deal with missing values in the dataset is one of its primary advantages. By finding the optimum way to impute missing data during the training process, the algorithm provides an internal mechanism for handling them. Instead of considering missing values as a separate category or applying imputation techniques, it automatically determines which branch of the tree to move to depend on the values that are missing. We worked on implementing the algorithm using the library xgboost, and the model was fit on the dataset similar to the one used in training the previous approaches.

4. Voting Regressor (XGBoost and Random Forest Regressor Model)

The Random Forest method and the XGBoost algorithm can be used to maximize their combined benefits and minimize their drawbacks. Their forecasts can be combined to improve performance overall and increase the precision of the outcome.

The capabilities of Random Forest include handling complex relationships, high-dimensional data, and reducing overfitting. Each tree in the ensemble of decision trees is trained using a different random subset of characteristics and samples. Random forests can provide reliable predictions and manage a variety of data circumstances by averaging the forecasts of several trees. We used the Voting Regressor algorithm of scikit learn to implement it.

3. Results

3.1 Arima Model

After removing the trend from the dataset, the next step involves applying the auto. arima algorithm to the transformed data. The purpose of this process is to automatically determine the optimal parameters for the ARIMA model. The auto. arima algorithm identifies the most suitable parameters for the ARIMA model as (3,0,2) shown in Figure 3.1.1, indicating three autoregressive terms (AR), zero differencing (I), and two moving average terms (MA). Furthermore, zero means suggests that the ARIMA model does not incorporate any additional means other than the inherent components captured by the AR, I, and MA terms. Zero means also indicates that the detrending has been successfully applied to the dataset.

```
arima_model <- auto.arima(log_diff_data$diff_Weekly_Sales)

print(arima_model)
```

```
Series: log_diff_data$diff_Weekly_Sales
ARIMA(3,0,2) with zero mean

Coefficients:
          ar1      ar2      ar3      ma1     ma2
      -0.3321  -0.7572  -0.5184  -0.5802  0.4143
s.e.   0.1247   0.0485   0.0995   0.1285  0.1210

sigma^2 = 0.007863:  log likelihood = 128.52
AIC=-245.05   AICc=-244.35   BIC=-227.98
```

Figure 3.1.1 ARIMA Model

Once the ARIMA model has been identified, the next step involves examining the residuals for normality, independence, and correlation. For the normality, the residuals are normally distributed as it follows a bell-shaped curve when looking at the histogram in Figure 3.1.2 and the QQ plot in Figure 3.1.2 shows that most points are approximately linear and lie close to the diagonal line, which indicates that the errors are normally distributed. By looking at the ACF and PACF plot of the residuals, the lags of the residuals fall within the 95% confidence interval of white noise. It means there is no significant residual autocorrelation remaining in the dataset. On the other hand, I perform the Box-Ljung test correlation test. The test checks whether the residuals are correlated. The null hypothesis is that the residuals are uncorrelated and the alternative hypothesis is that the residuals are correlated. After performing the Box-Ljung test, the p-value is 0.9799 which is larger than 0.05. It fails to reject the null hypothesis which indicates that the residuals are uncorrelated. And we also check the Box-Pierce test, the p-value is 0.9914 which is also larger than 0.05. It fails to reject the null hypothesis. Both tests determine the residuals are uncorrelated.

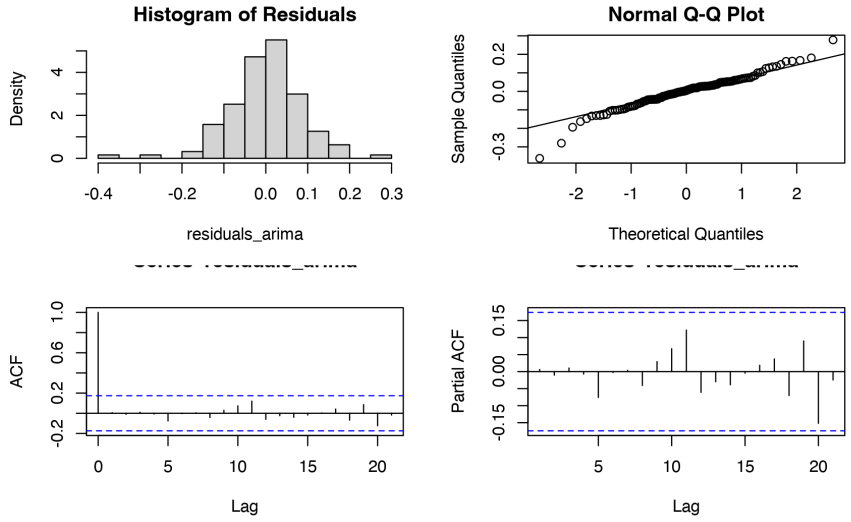

Figure 3.1.2 Residuals Checking

After conducting a thorough evaluation of the residuals, it can be concluded that the chosen ARIMA(3,0,2) model with zero means is suitable and satisfactory. I proceeded to make predictions for the future 15 timestamps every week, starting from 2012-11-01. In order the check the accuracy of the model, I compare the true values and the predicted values which are shown in Figure 3.1.3. From the zoomed-in plot, it is evident that the predicted values and the original values exhibit a high degree of similarity and alignment. The predicted values closely follow the pattern and fluctuations of the original values, indicating a strong match between the two.

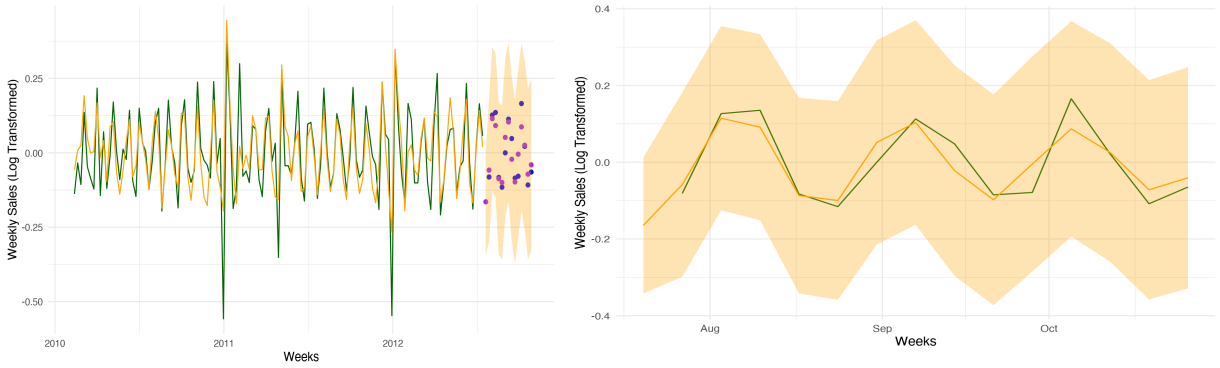

Figure 3.1.3 Transformed Weekly Sales True vs Predicted

After converting all the transformed data, including the predicted values, back to the original data scale, Figure 3.1.4 visualizes the comparison. From the zoom-in plot, we see the predicted values are a little shift from the original values. However, the predicted values are all within the 95% confidence interval. This suggests that the model captures the overall trend and variety, providing reliable predictions.

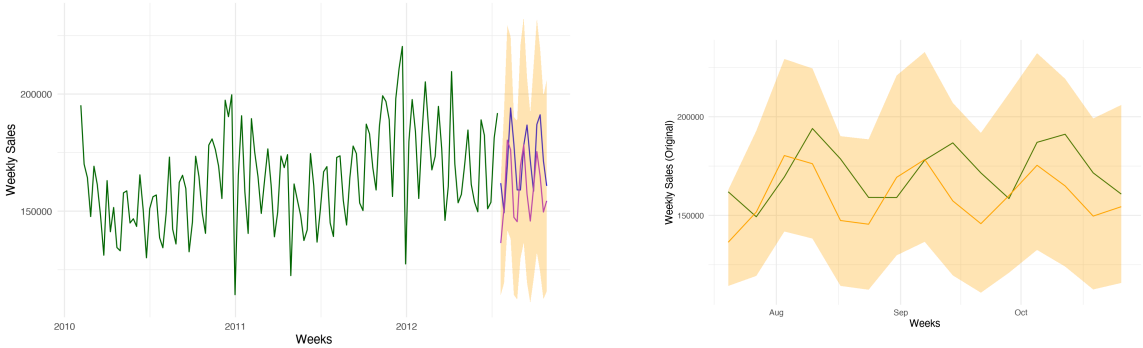

Figure 3.1.4 Original Weekly Sales True vs Predicted

Despite the ARIMA(3,0,2) model with zero means effectively capturing the overall trend and variability of the data, it is important to note that the Mean Absolute Error (MAE) for this model is calculated to be 12283.3. This value is considered quite high.

3.2 Simple Linear Regression

To comprehensively understand how the models performed in this dataset, we started with training the linear regression model with one predictor and the target variable of Weekly Sales. The predictor variable considered for analysis was Departments in Stores. We then cumulatively added features and evaluated the performance of the model, ultimately training the model on the entire dataset. Based on the results we were able to understand that a simple regression approach would not be able to capture the complexities in the dataset. We evaluated the model using Mean Absolute Error and Coefficient of determination scores. Figure 9 shows the scores obtained by the model.

```
In [31]: from sklearn.linear_model import LinearRegression
         from sklearn.model_selection import train_test_split
         from sklearn.metrics import mean_absolute_error, r2_score

         lgr = LinearRegression()
         x_lgr = train.drop('Weekly_Sales', axis=1).values
         y_lgr = train['Weekly_Sales'].values

         x_train_lgr, x_test_lgr, y_train_lgr, y_test_lgr = train_test_split(x_lgr, y_lgr, test_size=0.2, random_state=4)

         x_train_lgr = x_train_lgr.reshape(-1, x_lgr.shape[1])
         x_test_lgr = x_test_lgr.reshape(-1, x_lgr.shape[1])

         # x_test_lgr = x_test_lgr.reshape(-1,1)
         # x_train_lgr = x_train_lgr.reshape(-1,1)
         lgr.fit(x_train_lgr, y_train_lgr)

         ypred = lgr.predict(x_test_lgr)

         mae = mean_absolute_error(y_test_lgr, ypred)
         print('MAE of LGR :', mae)
         r2 = r2_score(y_test_lgr, ypred)
         print('R2 score of LGR :', r2)

         MAE of LGR : 14415.582197505957
         R2 score of LGR : 0.10413462397867557
```

Figure 3.2.1 MAE and Coefficient of Determination Scores of Linear Regression Model

3.3 Random Forest Modeling

The model performed considerably better than the Linear Regression model initially trained with the MAE score of 2514, and the Coefficient of Determination score of 0.94, indicating that the model was able to capture most of the complexities in the dataset. Figure 3.3.1 below shows the MAE performance of the model. Figure 3.3.2 below shows the Coefficient of determination score performance of the model Feature importances can be seen in Figure 3.3.3. Figure 3.3.4 shows the model performance. Based on the analysis it seemed that the Department in the Stores, followed by the Size of stores, and CPI were the key features that influenced the decision of the model.

```
In [25]: import pandas as pd
         from sklearn.ensemble import RandomForestRegressor
         from sklearn.metrics import mean_absolute_error, r2_score
         from sklearn.model_selection import train_test_split, GridSearchCV


         # Split the dataset into training and test sets
         X = train.drop('Weekly_Sales', axis=1)
         X_trans = mx.fit_transform(X)
         y = train['Weekly_Sales']
         X_train, X_test, y_train, y_test = train_test_split(X_trans, y, test_size=0.2, shuffle=False)

         # Create a Random Forest model
         model = RandomForestRegressor(random_state = 4, n_estimators=300)

         # Train the model
         model.fit(X_train, y_train)

         # Make predictions
         y_pred = model.predict(X_test)

         # Calculate mean absolute error (MAE)
         mae = mean_absolute_error(y_test, y_pred)
         print('MAE:', mae)

         MAE: 2514.72089648747
```

Figure 3.3.1 MAE of Random Forest Regression Model

```
In [26]: r_score = r2_score(y_test, y_pred)
         print(r_score)

         0.939947180495858
```

Figure 3.3.2 Coefficient of Determination Scores of Random Forest Regression Model

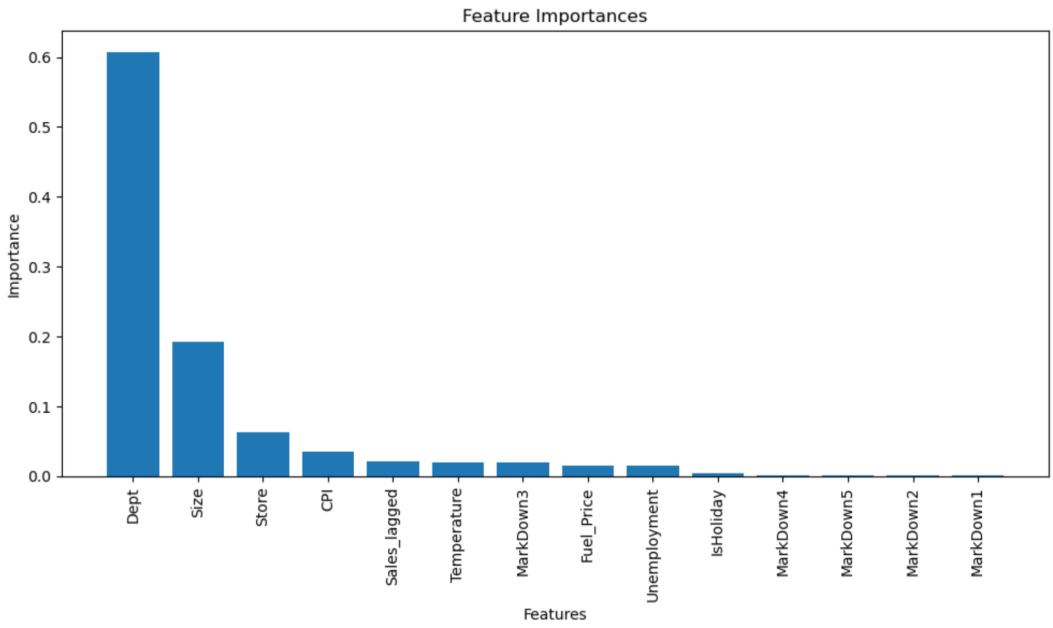

Figure 3.3.3 Feature importance graphs of the Random Forest Regression model

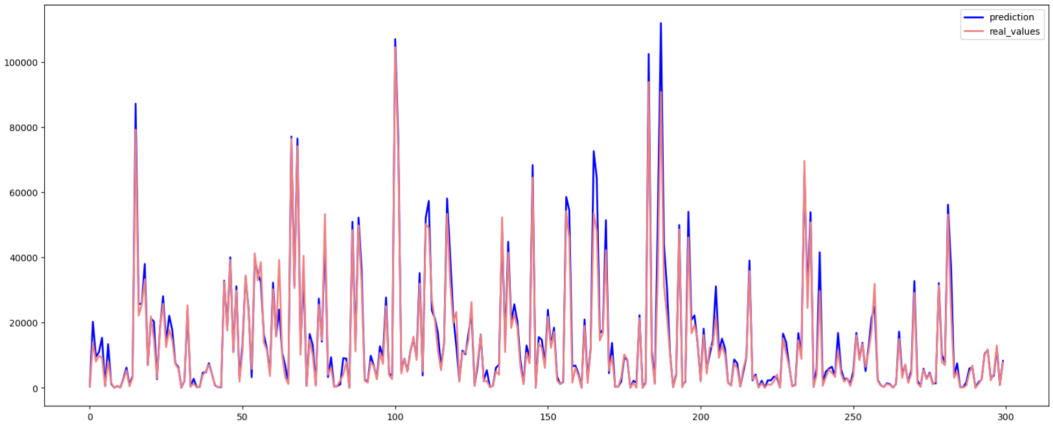

Figure 3.3.4 Predictions of Random Forest Regression Model

3.4 Extreme Gradient Boosting

The performance of the model was evaluated using MAE and Coefficient of determination scores. Additionally, we plotted the feature importances graph for the model to understand what features were key in influencing the predictions for the model. Figure 3.4.1 below shows the MAE scores for the model and the Coefficient of determination scores for the model. Figure 3.4.2 shows the prediction performance of the model. The model seemed to be closer to performance with Random Forest Algorithm with the MAE score of 3502, and Coefficient of determination scores of 92.4%. Figure 3.4.3 shows the feature importance, and we were able to find out that the Department, and Size of stores both were almost equally important for the XGBoost model, unlike Random Forest where Department was the main feature.

```
In [31]: import xgboost as xgb

         xgb = xgb.XGBRegressor(random_state=4, n_estimators=300)

         xgb.fit(X_train, y_train)

         # Make predictions
         y_pred = xgb.predict(X_test)

         # Calculate mean absolute error (MAE)
         mae = mean_absolute_error(y_test, y_pred)
         print('MAE of XGB :', mae)

         r_score = r2_score(y_test, y_pred)
         print(r_score)

         MAE of XGB : 3502.572126145665
         0.9247254850851936
```

Figure 3.4.1 MAE and Coefficient of Determination Scores of XGB Regression Model

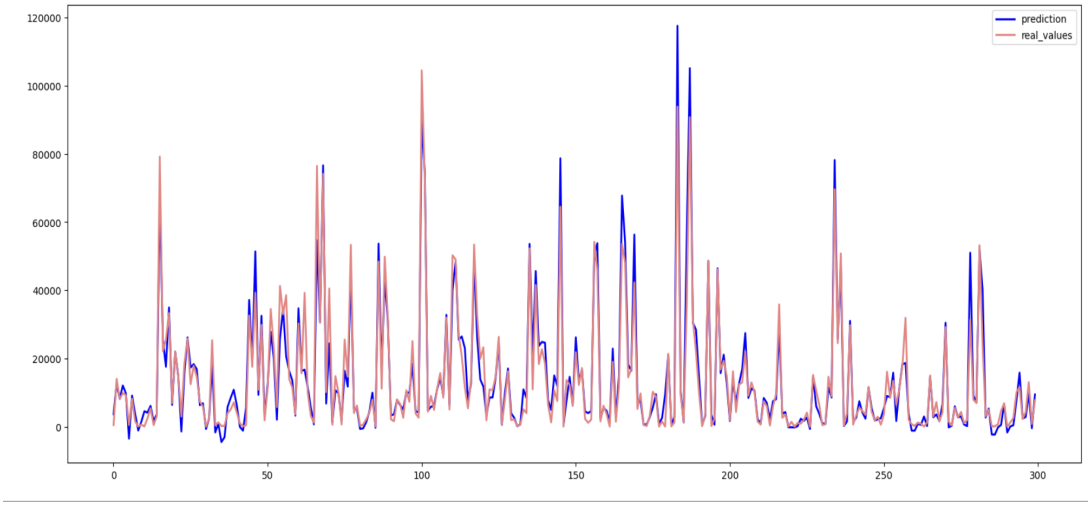

Figure 3.4.2 Predictions of the XGB Regression Model

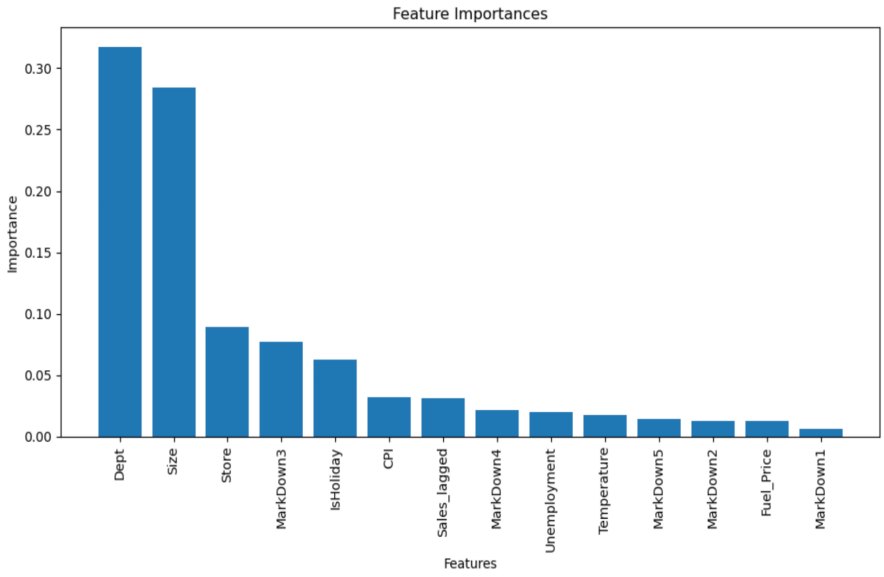

Figure 3.4.3 Feature Importance Graphs of XGB Regression Model

3.5 Voting Regressor (Random Forest Regressor + XGBoost Regressor)

The ensemble model was evaluated using MAE and Coefficient of determination score values. Based on the results obtained, it seemed to perform as well as the Random Forest Model with the Coefficient of Determination score of 94%, however, the MAE score was 2803, slightly higher than the MAE obtained by the Random Forest model. Figure 16 below shows the MAE and Coefficient of determination scores. Figure 3.5.2 shows the prediction plots.

```
In [35]: from sklearn.ensemble import VotingRegressor

ensemble_model = VotingRegressor([('rf', model), ('xgb', xgb)])

ensemble_model.fit(X_train, y_train)

y_pred = ensemble_model.predict(X_test)  # X_test represents your test data features


mae = mean_absolute_error(y_test, y_pred)
print('MAE of Ensemble between RandomForest and XGB :', mae)

r_score = r2_score(y_test, y_pred)
print(r_score)
```

```
MAE of Ensemble between RandomForest and XGB : 2803.9407289824167
0.9413685381049289
```

Figure 3.5.1 MAE and Coefficient of Determination Scores of Voting Regression Model

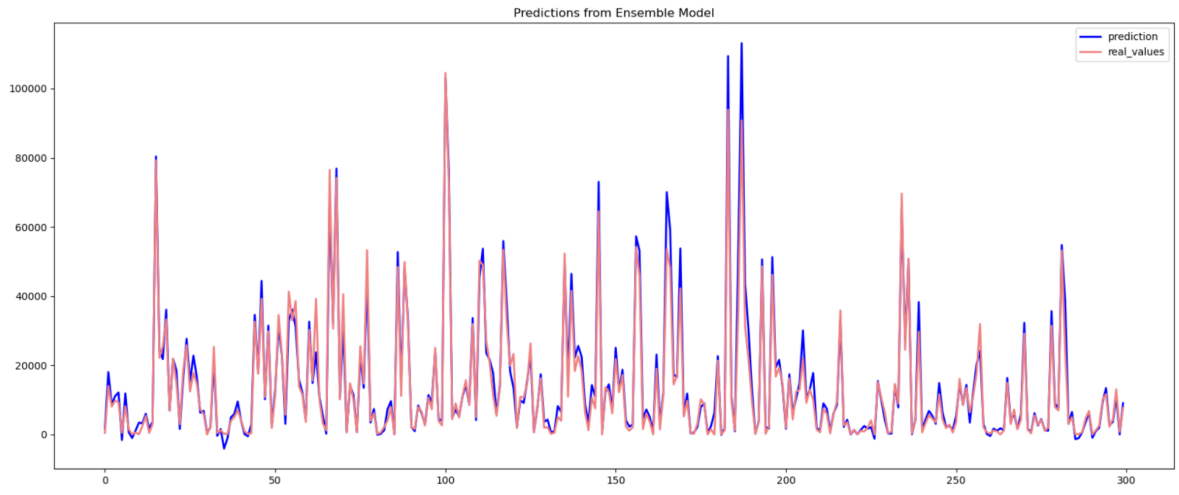

Figure 3.5.2 Predictions of Voting Regressor Model

4. Discussion

This study looked at the effects of things like weather, the CPI, and holidays on sales across 45 shops and roughly 199 departments to forecast the weekly sales of Walmart data using lagged variables. With a ratio of 1.07 between holiday and non-holiday sales, exploratory data analysis and machine learning algorithms revealed that although holidays looked to contribute to increased sales, the total influence was not statistically significant. Sales were generally higher during the holidays, according to the initial exploratory data analysis, suggesting a possible beneficial influence. However, a closer look revealed that the ratio of holiday to non-holiday sales suggested that the holidays themselves might not be the main cause of higher sales. Sales were generally higher during the holidays, according to the initial exploratory data analysis, suggesting a possible beneficial influence. However, a closer look revealed that the ratio of holiday to non-holiday sales suggested that the holidays themselves might not be the main cause of higher sales. This observation was further verified by machine learning algorithms, which repeatedly corroborated the finding that, when lag variables were taken into account, the impact of holidays on weekly sales was not statistically significant.

We examined the dataset's distribution to choose the best modeling approaches and found that the skewed distribution of the feature columns made tree-based regression algorithms like Random Forest and XGBoost models ideal candidates. We improved our understanding of how the models projected sales based on past data by adding lagged variables to the dataset. It should be noted that the ARIMA model with a lag order of 3 performed better in predicting sales, which affected the choice of the lagged variables. The performance measures, such as Mean Absolute Error (MAE) and Coefficient of Determination, were further improved by hyperparameter adjustment of the estimators in the tree-based algorithms.

With a high Coefficient of Determination score of 94%, the Random Forest model consistently beat the other examined models. This rating shows that the Random Forest model successfully captured the dataset's complexity, giving a thorough insight into the variables affecting weekly sales. Because sales were recorded by departments inside each store, our early suspicions about duplication in the dataset were founded. The sales were correctly recorded, and we used pandas to validate that there were no duplicates. Furthermore, we saw that the departments' and stores' performances affected the ARIMA model's ability to provide higher MAE scores. The machine learning models, which used the entire dataset for prediction, did not operate in this way.

One major drawback of this study is the dataset's short timeline and duration, which might make it difficult to do a thorough examination of the topic. Despite our confidence in the generalizability of the employed techniques, it may not be acceptable to base analysis simply on this dataset. further recent data should be included to get over this restriction and boost our study even further. By including recent data, we can gain new perspectives, strengthen the validity and trustworthiness of our conclusions, and increase the study's overall robustness.

In the future, it would be beneficial to expand the models' predicting skills beyond a three-week window. Further research into the application of deep learning algorithms like Long Short-Term Memory (LSTM) and Recurrent Neural Networks (RNNs) may yield new information and improve the precision of sales forecasts. Future studies could also benefit from looking into the potential for developing store-specific models to offer individualized recommendations for certain stores.

Incorporating lag variables and examining the effects of many parameters, this study, in conclusion, helps us understand how to anticipate weekly sales at Walmart. The results indicate that although holidays may help to improve sales, their total impact is not statistically significant when lag effects are taken into account. The Random Forest model in particular, which was adopted, worked well in capturing the complexity of the dataset. To improve the precision and applicability of sales predictions in the retail sector, it is crucial to take into account the dataset's constraints and look into potential new study areas.

5. References

Adetunji, Abigail & Funmilola Alaba, Ajala & Ajala, & Oyewo, Ololade & Akande, Yetunde & Oluwadara, Gbenle & OLUWATOBI, AKANDE. (2022). House Price Prediction using Random Forest Machine Learning Technique. Procedia Computer Science. 199. 10.1016/j.procs.2022.01.100.

Chen, Tianqi, and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System." *arXiv.Org*, 10 June 2016, arxiv.org/abs/1603.02754.

Cukierski, Will. *Walmart Recruiting - Store Sales Forecasting*, 2014, www.kaggle.com/competitions/walmart-recruiting-store-sales-forecasting.

Gupta, Aayush. "Walmart Recruiting-Store Sales Forecasting." *Medium*, 28 July 2021, medium.com/geekculture/walmart-recruiting-store-sales-forecasting-b8b2f4cf19b1.

Jeswani, Rashmi. "Predicting Walmart Sales, Exploratory Data Analysis, and Walmart Sales Dashboard: Ischool Projects." *RIT*, 1 Dec. 2021, www.rit.edu/ischoolprojects/node/104009.

Kuzlu, M., Cali, U., Sharma, V., & Güler, Ö. (2020). Gaining insight into solar photovoltaic power generation forecasting utilizing explainable artificial intelligence tools. IEEE Access, 8, 187814-187823.

Santaella Colón, José Gil. "Data Mining Techniques and Machine Learning Model for Walmart Weekly Sales Forecast." *PRCR Principal*, 1 Jan. 1970, prcrepository.org/xmlui/handle/20.500.12475/174.

"Sklearn.Ensemble.Votingregressor." *Scikit*, scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingRegressor.html. Accessed 2 June 2023.

"Walmart." *Wikipedia*, 1 June 2023, en.wikipedia.org/wiki/Walmart.

Xue, Liang, et al. "A Data-Driven Shale Gas Production Forecasting Method Based on the Multi-Objective Random Forest Regression." *Ministry of Education - Saudi Arabia*, ksascholar.dri.sa/en/publications/a-data-driven-shale-gas-production-forecasting-method-based-on-th-4. Accessed 2 June 2023.

Yi, Siming. "Walmart Sales Prediction Based on Machine Learning." *Highlights in Science, Engineering, and Technology*, vol. 47, 2023, pp. 87–94, https://doi.org/10.54097/hset.v47i.8170.