Online Retail Forecasting Project Technical Documentation

Part 1: Introduction & Business Problem

The Online Retail Forecasting Project is designed to use the power of data analytics to enhance the operational efficiency and strategic planning of online retail businesses in the UK. By analyzing patterns, trends, and seasonal variations in past transactions and weather data, we aim to **predict future daily sales in the UK**. This involves a deep dive into past transactions to identify patterns, trends, and seasonal fluctuations that significantly affect sales outcomes.

In this context, the project will employ a range of advanced predictive modelling techniques, **including both statistical and machine learning models**. The objective is to develop robust forecasting models that can reliably anticipate daily sales figures. These models are expected to serve as critical tools for inventory managers, enabling them to optimize stock levels based on predicted sales volumes, thus reducing the cost of overstocking and minimizing the risk of stockouts. In addition, integrating weather data acknowledges the significant impact of weather conditions on consumer online buying patterns, enabling a more contextually relevant forecast.

The outcomes of this project are anticipated to have a significant impact on the business's bottom line. By providing accurate sales forecasts, the project will empower decision-makers with the insights needed to make informed strategic choices. These choices will span various aspects of the business, from inventory management and marketing to financial planning and customer service, ultimately enhancing the overall competitiveness and market responsiveness of the online retail business.

Part 2: Data, Data Preprocessing, EDA

- Data

The data is downloaded directly from the UCI Machine Learning Repository (https://archive.ics.uci.edu/dataset/502/online+retail+ii), which contains transaction data between 01/12/2009 and 09/12/2011 from a UK-based online retail.

The original dataset contains the following features:

Feature Name	Type	Description
Invoice	Nominal	Each transaction is given a unique 6-digit identifier. If it starts with 'C', it marks a cancellation.
StockCode	Nominal	Each unique product is given a distinct 5-digit integer as its identifier.
Description	Nominal	The description of the product.
Quantity	Numeric	The quantities of each product per transaction. The quantities associated with Invoice starting with 'C' are negative.
InvoiceDate	Numeric	The date and time at which a transaction was recorded.
Price	Numeric	Product price per unit.

Feature Name	Type	Description
Customer ID	Nominal	Each customer is given a unique 5-digit integer as their identifier.
Country	Nominal	The name of the country where a customer resides.

It was observed that rows with negative quantities correspond to invoices prefixed with 'C', indicating cancellations. These will be adjusted to achieve net zero sales.

- Data Pre-processing

The preprocessing phase for this project involves a series of steps to prepare the dataset for analysis.

Firstly, we concatenated two spreadsheets from the downloaded Excel file to form the whole dataset with time spans from 01/12/2009 to 09/12/2011. We then created a new `Sales` column by multiplying `Quantity` with `Price`. We also derived the `Date` column from `Invoice` column to capture only the date on which a transaction was recorded.

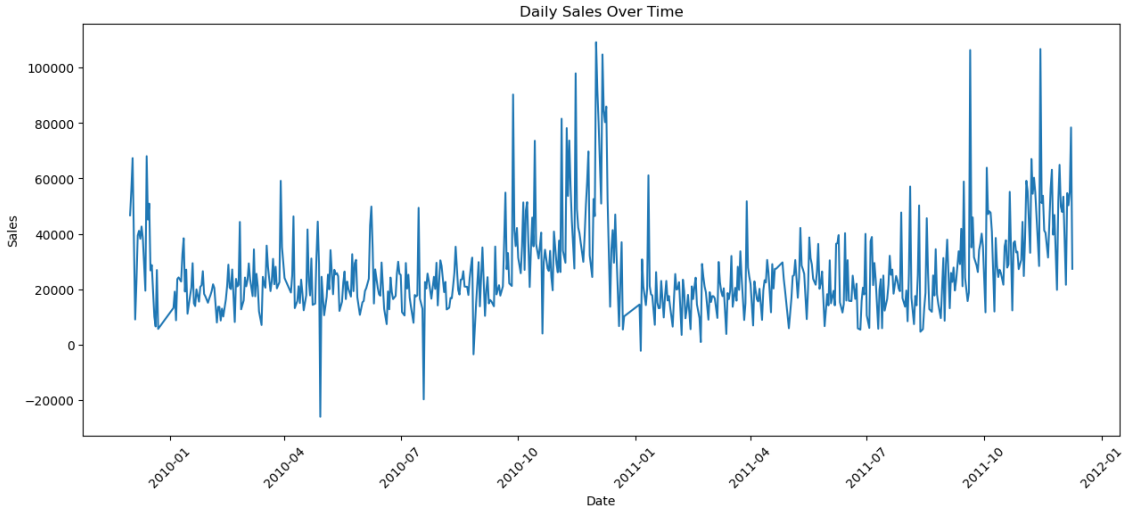
Secondly, to get daily sales, we aggregated the sales and formed the **daily_sales** dataframe that spans from 01/12/2009 and 09/12/2011. We derived the `Year` , `Month` , and `Week` column from `Date` column. Hence, there are five features: `Date` , `Year` , `Month` , `Week` , and `Sales` in the **daily_sales** dataframe which will be our primary data for forecasting daily sales. Each row in this dataframe represents the total daily sales for that day, and we have daily sales data for 604 days.

Till this step, the **daily_sales** dataframe has the following features:

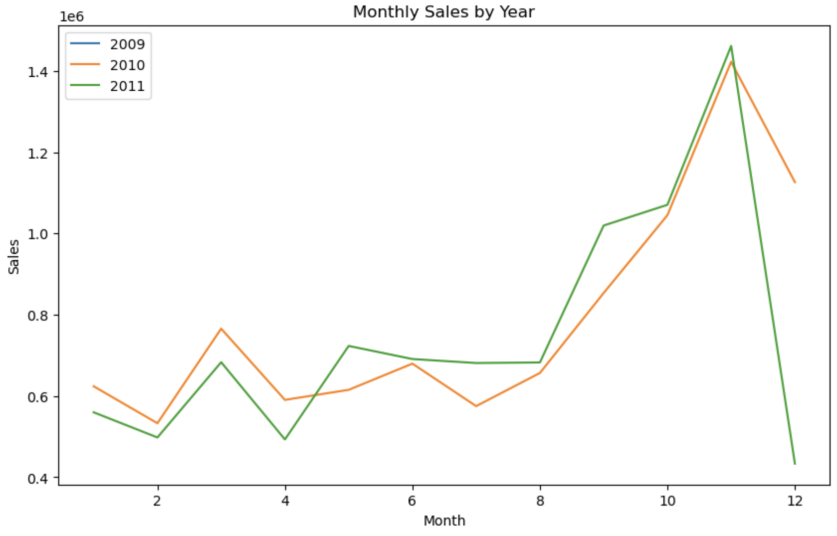
Features	Type	Description
Date	Numeric	The date on which a transaction was recorded.
Sales	Numeric	Sales per transaction (Quantity * Price).
Month	Numeric	Month (1-12)
Year	Numeric	Year (2009-2011)
Weekday	Nominal	Weekday (Monday-Friday)

- Exploratory Data Analysis

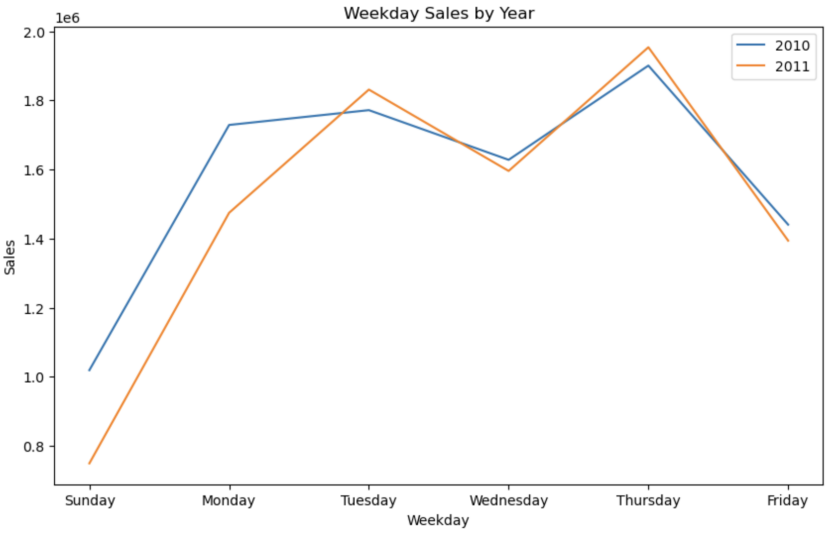
The graph “Daily Sales over Time” shows fluctuations in daily sales, with some days reaching as high as approximately 120,000, while other days show negative sales. There are noticeable spikes in sales between October to January. The overall trend appears to be somewhat cyclical. This could imply seasonality in purchasing patterns, with potential peaks around the same time each year, likely around holidays or sale periods.



The chart "Monthly Sales by Year" shows a comparison of monthly sales for the years 2010 and 2011, as indicated by the two colored lines on the graph. The absence of a line for 2009 is due to the dataset only containing data for a single month from that year. Monthly sales for both years resemble each other with a moderate increase from the beginning of the year to a significant rise towards the end of the year, peaking in November. Following this peak, there is a drop in December. One thing to note is that our data ends on December 9th, 2011 and that's why there is a sharp decrease in the green line in December. However, if we have enough data for the month, we would expect a similar sales pattern for these two years.



The chart "Monthly Sales by Year" compares sales for each weekday over two different years, 2010 and 2011. For both years, the graph exhibits a similar pattern: sales start lower on Sunday, then there is a sharp increase on Monday, reaching a peak on Thursday. Sales remain relatively high on Tuesday and then show some fluctuation mid-week. The lines for 2010 (blue) and 2011 (orange) track closely together, suggesting consistent sales patterns across the two years. One thing to notice is that there is no data for Saturday.

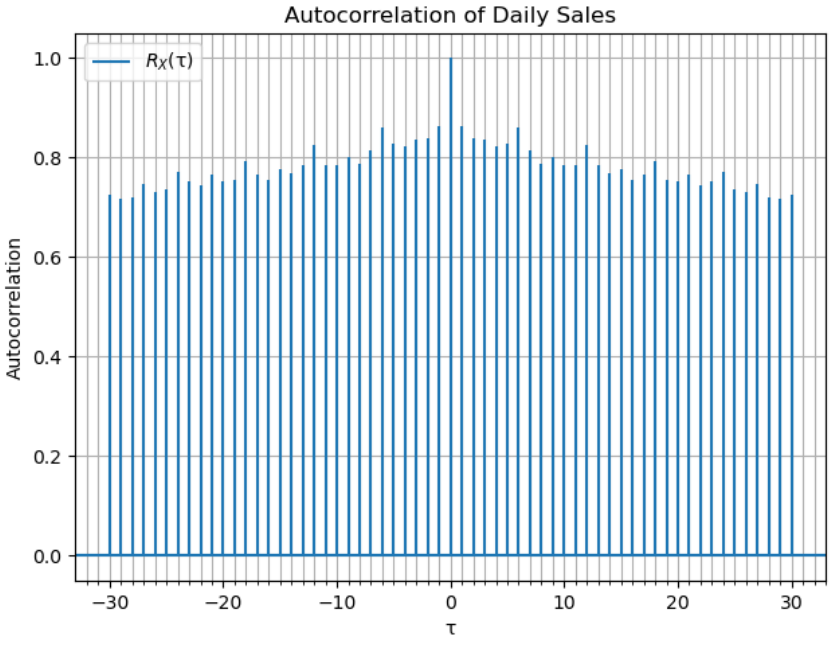


- Pre-Modeling EDA

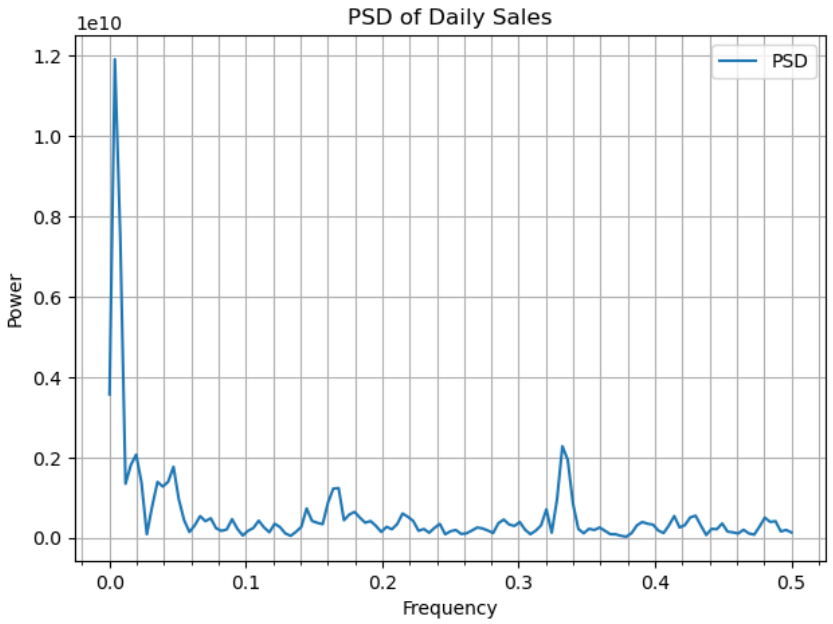
1. Check for Seasonality and Trend

Based on our exploratory data analysis (EDA), we suspect that our data is non-stationary. To confirm this hypothesis, we have incorporated autocorrelation, power spectral density (PSD), and seasonal decomposition into our analysis.

The autocorrelation graph indicates a strong seasonal pattern as the autocorrelation coefficients are significant at multiple lags and the peaks are at regular intervals.



The PSD plot shows a spike at a very low frequency, corresponding to a long-term trend or cycle.



The four plots below indicate a seasonal decomposition of daily sales into three main components: trend, seasonality, and residual. From the trend plot, we can see that there is an upward trend towards the end of the year. From the seasonality plot, we can see that there are regular fluctuations that seem to repeat at consistent intervals. This suggests a strong seasonal pattern in the daily sales. From the residual graph, we see that most of the residuals are close to zero, which is good. However, there are a few outliers in October/November of both years, some of which are quite significant.

2. Convert Daily Sales to Stationary

To substantiate our hypothesis with statistical evidence, we conducted an Augmented Dickey–Fuller (ADF) test on our original daily sales data. The ADF test aims to determine the presence of a unit root in the time series, a sign of non-stationarity. The test yielded a p-value of 0.045338. At a 1% significance level, we can not reject the null hypothesis, suggesting that the data is non-stationary.

Hence, we used the differencing technique to make the data stationary. We then run a ADF test again, and got a p-value of 0.000000 which is less than the 1% significant level. We can then reject the null hypothesis and conclude that our data is stationary.

	Original Data	After Differencing
p-value	0.045338	0.000000

This step is crucial as many time series forecasting methods assume stationarity, such as MA and AR models.

Part 3: Adding Relevant Features & Feature Engineering

- Adding Relevant Features

Many more features could affect online retail sales in the UK. For example, during holidays, more people would shop for gifts; during bad weather, more people would stay in and shop online. Hence, we added features like `Holiday` , `mean-temp` , `precipitation` , `snow_depth` . We got the UK holidays directly from the holiday library. We got the weather data from a Kaggle dataset (https://www.kaggle.com/datasets/emmanuelfwerr/london-weather-data). Since we can not find weather data in the UK, we used the weather data in London.

We also included several time-based features, including moving average features on Sales. They are `Sales_ma_3` , `Sales_ma_7` , `Sales_ma_15` , `Sales_ma_30` .

- Feature Engineering

As seen above in the Data Preprocessing section, weekday is not numerical. Weekdays have a natural cyclical nature. We did Sin/Cos transformations of weekdays to avoid ordinal relationships if encoding it using other methods. Added features include `Weeday_sin` , `Weeday_cos` .

The final **daily_sales** data frame has the following features:

Features	Type	Description
Date	Numeric	The date on which a transaction was recorded.
Sales	Numeric	Sales per transaction (Quantity * Price).
Month	Numeric	Month (1-12)
Year	Numeric	Year (2009-2011)
Weekday_sin	Numeric	Sin transformation of weekday
Weekday_cos	Numeric	Cos transformation of weekday
Holiday	Numeric	1 represents a holiday; 0 represents no holiday
Sales_ma_3	Numeric	3-day moving average of sales
Sales_ma_7	Numeric	7-day moving average of sales
Sales_ma_15	Numeric	15-day moving average of sales
Sales_ma_30	Numeric	30-day moving average of sales
mean_temp	Numeric	Mean temperature in London
precipitation	Numeric	Precipitation in London
snow_depth	Numeric	Depth of snow

Part 4: Target & Predictive Variables

The target variable is daily sales `Sales` .

Time-Series Models: The predictive variable for AR is simply historical daily sales data. The predictive variable for MA is historical daily sales errors. The ARIMA combines both historical daily sales data and historical daily sales errors. On top of ARIMA, SARIMA adds additional seasonal terms. The predictive variables for Facebook Prophet are every variable in daily_sales except the target variable.

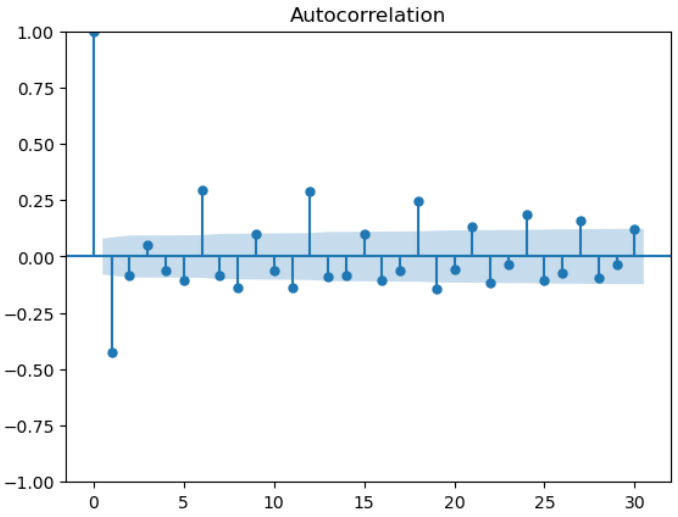
Machine Learning Models: The predictive variables for Elastic-Net, Random Forest, SVR, and XGBoost are every variable in daily_sales except the target variable and Date.

Part 5: Time Series Models

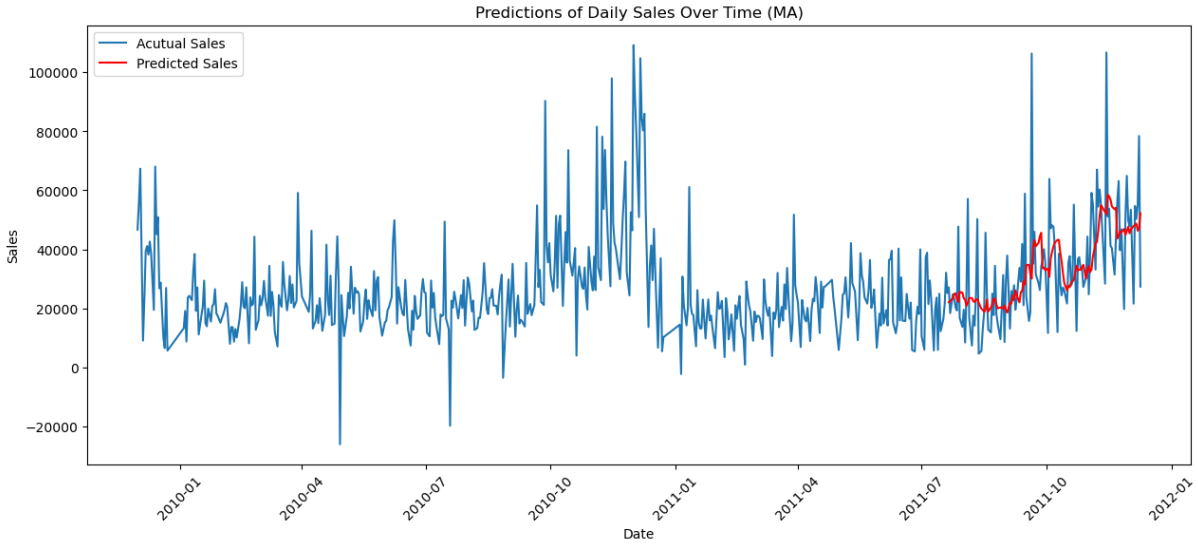
We will consider using statistical time-series models and machine-learning models to make the predictions. Some of the time-series models we explored are MA, AR, ARIMA, and SARIMA. To evaluate the performance of the time-series models, we split the data into 80% train and 20% test, and we will use MAPE(Mean absolute percentage error) as our model performance metric to compare different models.

Model 1: MA

The Moving Average (MA) is a fundamental technique in time series analysis, primarily used for smoothing data and highlighting trends by averaging past observations. We first analyzed the autocorrelation function (ACF) of the data, which can be seen below:

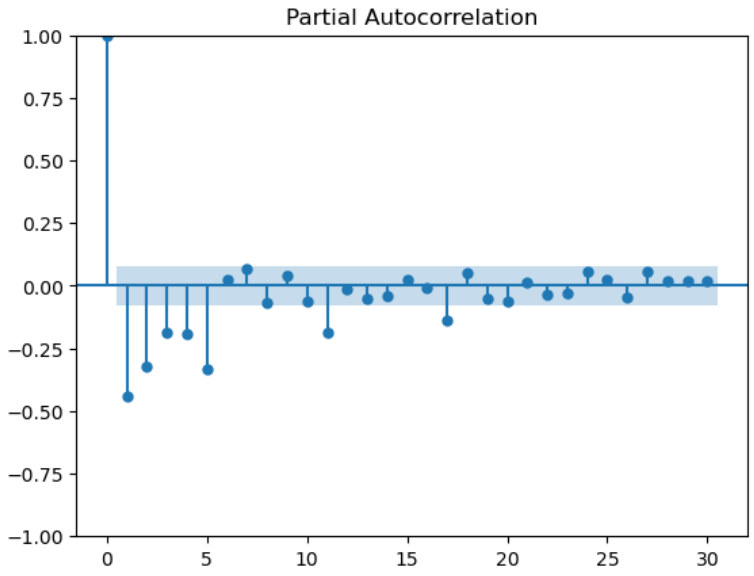


Upon examining the autocorrelation function (ACF) of daily sales, a distinct cut-off emerged at lag 6, guiding our choice of q = 6 for the MA model's order. This means that the forecast for today would be based on a combination of the errors from the last six days. The graph below shows how our model's forecasts compare to the actual test data. The model had a **test MAPE 0.4459**.

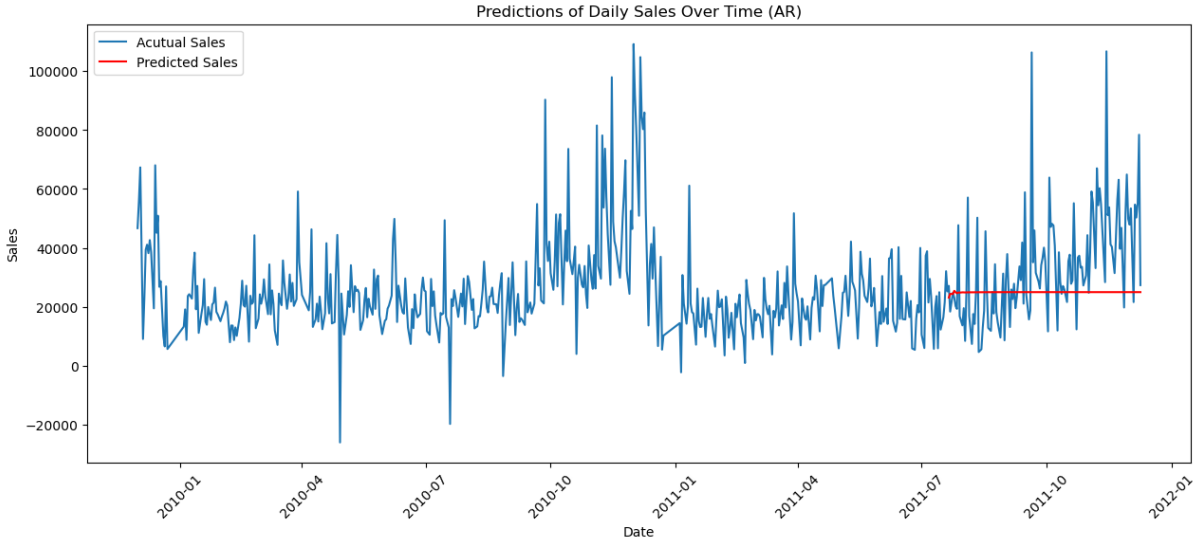


Model 2: AR

We also explored the Autoregressive (AR) model for our time series forecasting efforts. To identify the optimal order *p* for the model, we first analyzed the partial autocorrelation function (PACF) of the data, which can be seen below:



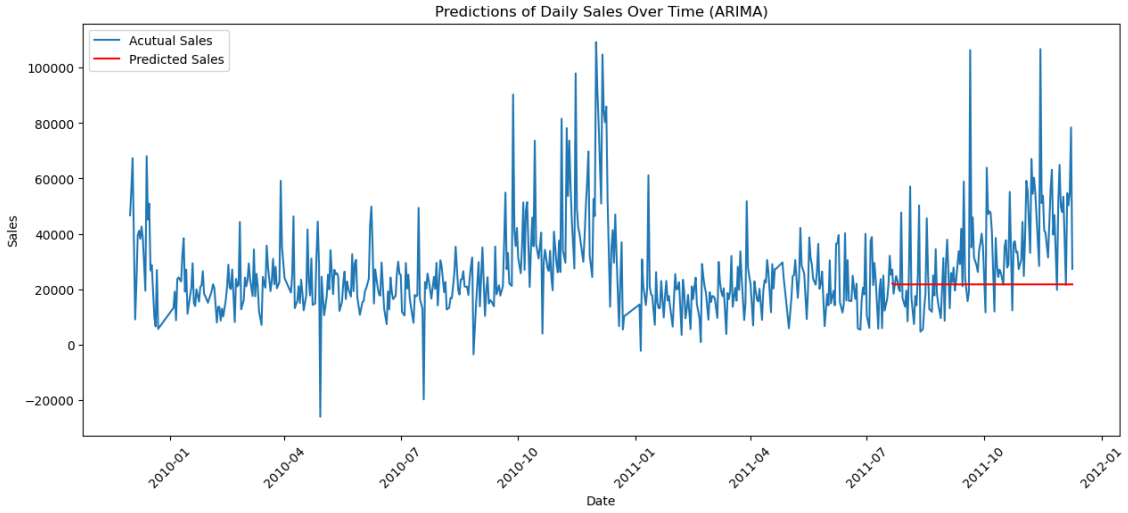
Upon examining the partial autocorrelation function (PACF) of daily sales, a distinct cut-off emerged at lag 5, guiding our choice of p = 5 for the AR model's order. The model had a **test MAPE 0.4986**.



Model 3: ARIMA

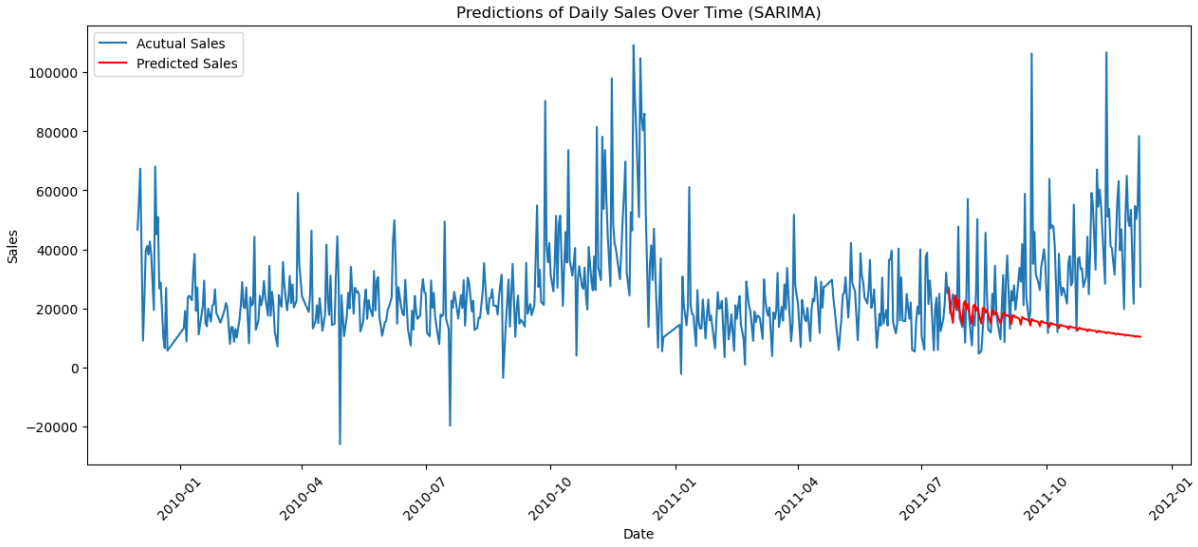
ARIMA **i**ncorporates an integration component (I) that allows for differencing the data, making it possible to model non-stationary time series that have a trend, which adds flexibility and capability to the model. It combines AR and MA components, enabling it to capture a broader range of autocorrelation structures in the time series data. We used the function auto_arima to find the best parameters (`p` , `d` , `q`) for ARIMA. The ARIMA(0, 1, 2) model achieved a **test MAPE of 0.4864**.

However, based on the visualization, the ARIMA model does not perform well in forecasting daily sales. The graph below shows how our model's forecasts compare to the actual test data.



Model 4: SARIMA

SARIMA extends ARIMA by adding a seasonal component to the model, making it more suitable for time series with strong seasonal effects. Our data does have a seasonal pattern, hence we would expect the SARIMA model to achieve better performance. We used the function auto_arima to find the best parameters (`p`, `d`, `q`, `P`, `D`, `Q`) for ARIMA. In our case, the SARIMA(5,0,2)(1,0,1)[6] achieved a **test MAPE of 0.5571**. The graph shows how our model's forecasts compare to the actual test data.

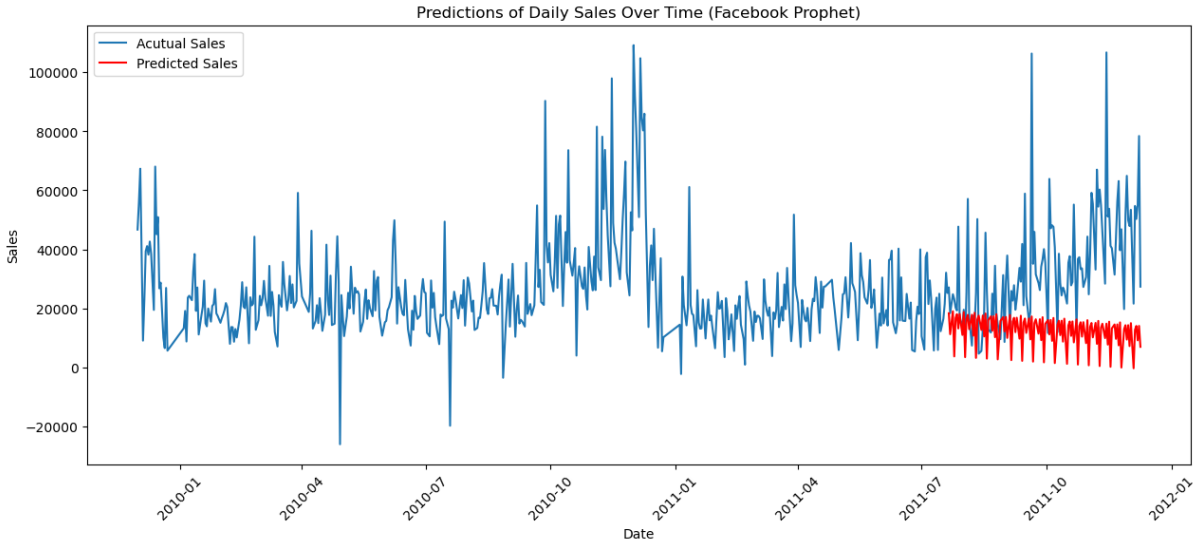


Model 5: Facebook Prophet

We also employed the Facebook Prophet model, a robust tool designed for predicting time series data with strong seasonal patterns. This model is favored for its ability to handle large datasets and its flexibility in accommodating holidays and other recurring events that significantly impact sales.

Upon applying the Facebook Prophet model to our dataset, we conducted a thorough evaluation of its performance metrics, with a specific focus on the Mean Absolute Percentage Error (MAPE). The

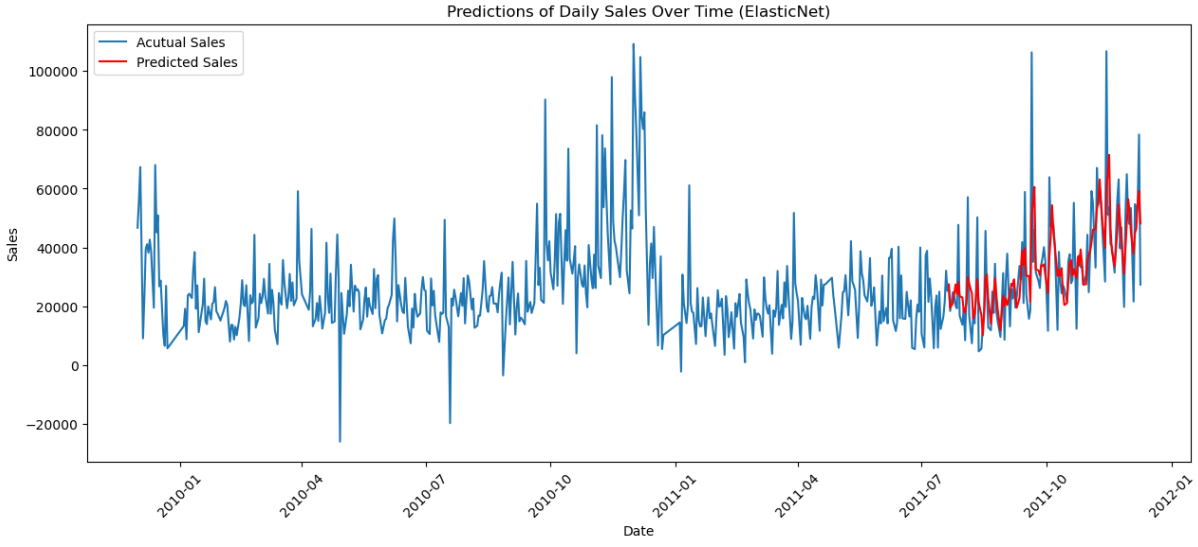
test MAPE achieved was 0.6294, which serves as a quantitative measure of the model's prediction accuracy. The graph shows how our model's forecasts compare to the actual test data.



Part 6: Machine Learning Models

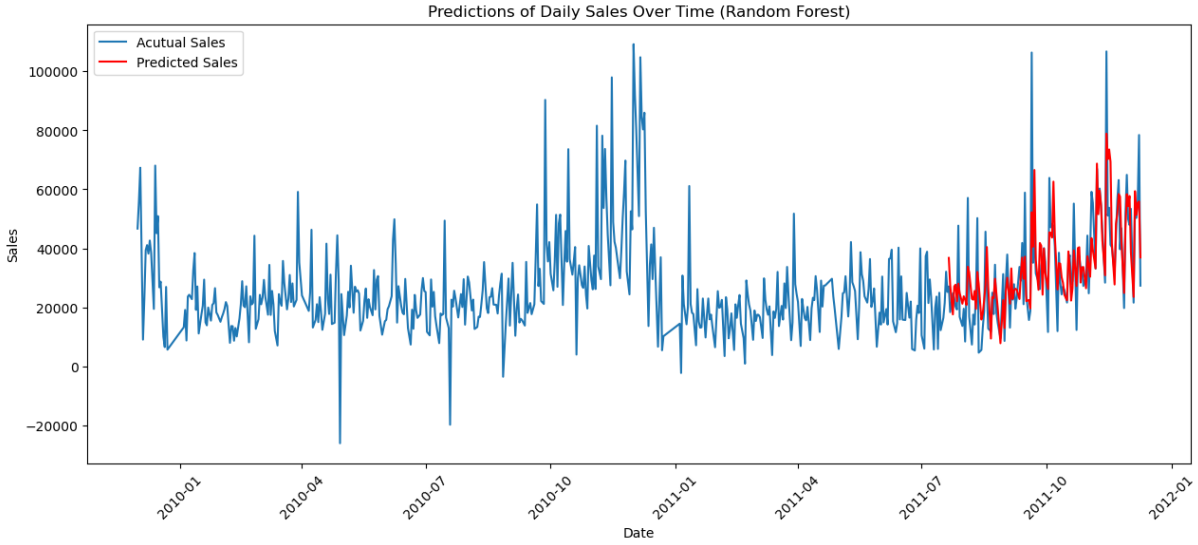
Model 6: Elastic Net Regularization

For machine learning models, we first fit Elastic Net Regularization, and get a **test MAPE of 0.3349**, indicating that the model's predictions were, on average, 33.49% off from the actual sales figures. This performance suggests that Elastic Net, a regularization technique combining L1 and L2 penalties, has provided a more accurate and generalizable predictive model for our sales data compared to any statistical time-series model before . The lower MAPE means a substantial improvement in forecast accuracy, reflecting the effectiveness of Elastic Net in handling feature selection and preventing overfitting. The graph below shows how our model's forecasts compare to the actual test data.



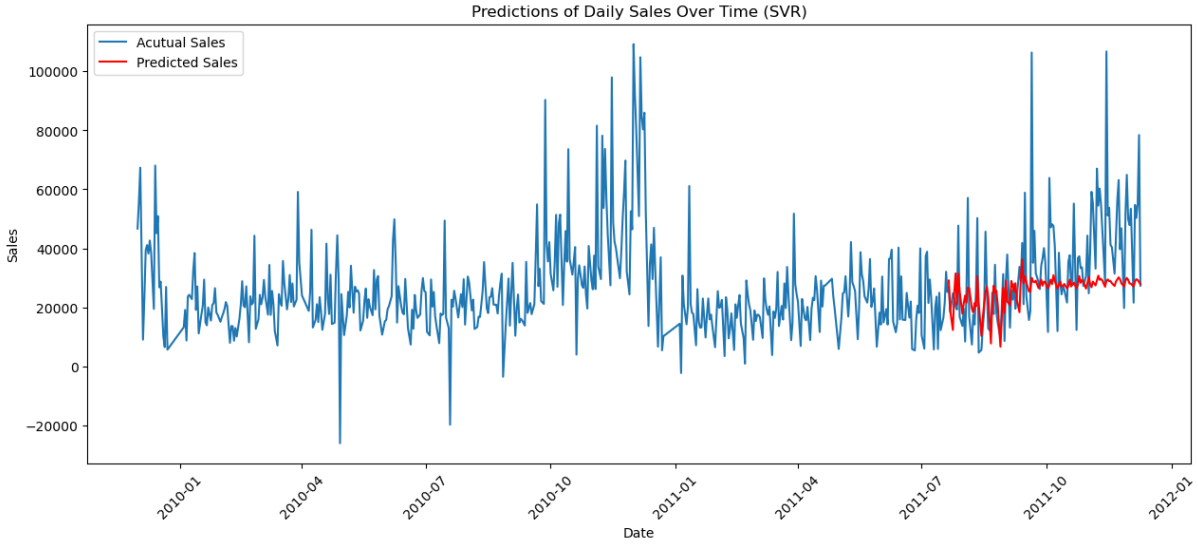
Model 7: Random Forest

Using the Random Forest algorithm, we achieved a **test MAPE of 0.3181**, which indicates the model's predictions were 31.81% off from the actual sales figures. This result shows that the Random Forest model performed slightly better than Elastic Net Regularization in forecasting accuracy, demonstrating its effectiveness in capturing complex patterns and relationships in the sales data. The graph below shows how our model's forecasts compare to the actual test data.



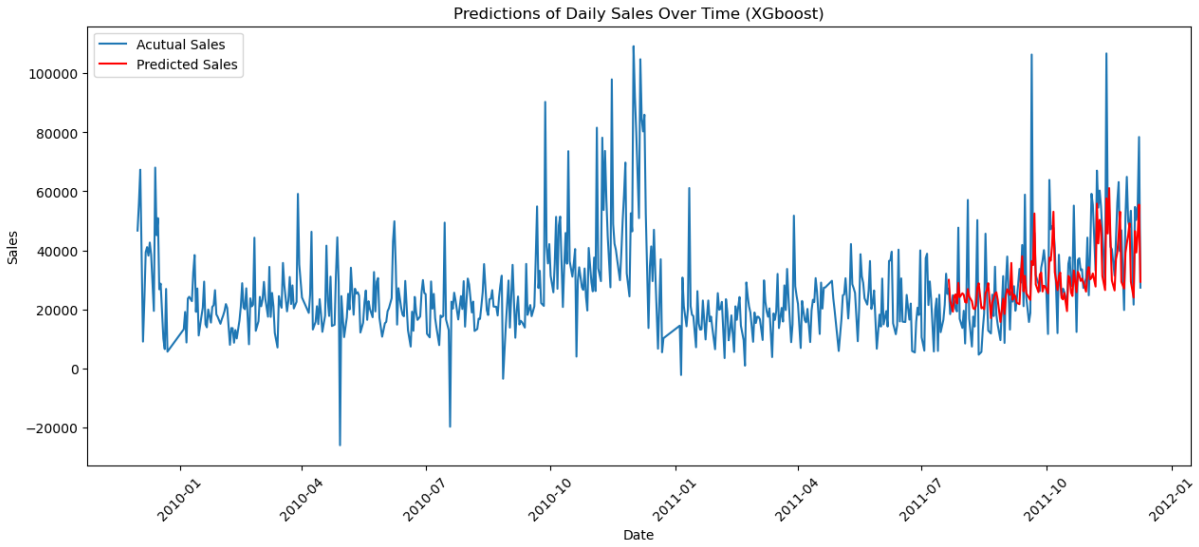
Model 8: SVR

Using the Support Vector Regression (SVR) model, we obtained a **test MAPE of 0.3901**. This result means the predictions were on average 39.01% different from the actual sales figures, showing that the SVR model had less predictive accuracy than both the Elastic Net Regularization and Random Forest models in this instance. In the graph below, we can see that the SVR can't capture the more cyclic pattern in the later part of the test data.



Model 9: XGBoost

Using the XGBoost model, the **test MAPE achieved was 0.3701**, which indicates the model's predictions were 37.01% off from the actual sales figures. This suggests that while the XGBoost model was more accurate than the SVR model, it was not as precise as the Random Forest or Elastic Net models in this particular forecasting task.



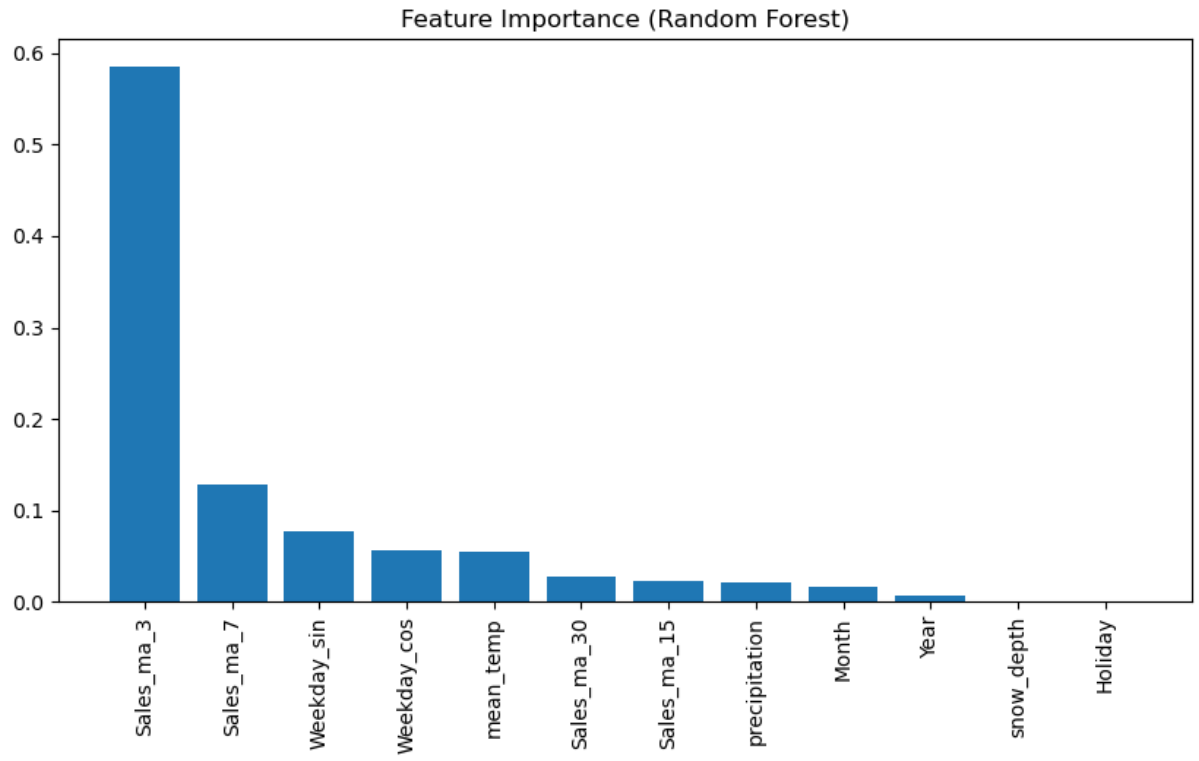
Part 7: Model Performance Comparison

According to the chart below, the Random Forest model has the lowest test MAPE of 0.3181, which means it is the best-performing model for predicting the daily online sales in the UK.

Model	Best Hyper-parameter	Best Test MAPE	Period 1 Test MAPE	Period 2 Test MAPE	Period 3 Test MAPE	Period 4 Test MAPE
MA	{'q': 6}	0.4459	0.6382	0.4410	0.3838	0.3535
AR	{'p': 5}	0.4986	0.7545	0.4049	0.3659	0.4622
ARIMA	{'p':0, 'd':1, 'q':2}	0.4864	0.6259	0.3919	0.4116	0.5151
SARIMA	{'p':5, 'd':0, 'q':2, 'P':1, 'D':0, 'Q':1}	0.5571	0.4747	0.4418	0.5613	0.7444
Facebook Prophet	{'changepoint_prior_scale': 0.001, 'seasonality_prior_scale': 10}	0.6294	0.5628	0.5268	0.5794	0.6852
Elastic Net Regression	{'alpha': 0.1, 'l1_ratio': 0.9}	0.3349	0.5410	0.4216	0.4236	0.5060
Random Forest	{'max_depth': None, 'min_samples_split': 2, 'n_estimators': 10}	0.3181	0.5647	0.2909	0.2496	0.1833
SVR	{'C': 10000, 'epsilon': 0.01, 'gamma': 0.5}	0.3901	0.8476	0.4233	0.3513	0.4330
XGBoost	{'learning_rate': 0.02, 'max_depth': 8, 'n_estimators': 50}	0.3701	0.6829	0.3022	0.2795	0.2207

Part 8: Best Performing Model (RF) Analysis

- Feature Importance Analysis



This feature importance plot from a Random Forest model shows the relative importance of different variables used to forecast sales. The most influential feature is **sales_ma3** and **sales_ma7**, which represents the average sales from three or seven days prior. The next important features are **weekday_sin** and **weekday_cos**, which are engineered features likely represents the cyclical nature of weekdays.

Mean_temp suggests that temperature has a moderate impact on sales, implying a possible relationship between weather conditions and consumer buying behavior. **Sales_ma30** and **sales_ma15** are moving averages over 30 and 15 days, respectively, indicating the role of relatively long-term sales trends in predicting future sales.

Features like **precipitation**, **month**, **year**, **snow_depth**, and **holiday** have relatively lower importance scores. **Precipitation** and **snow_depth** relate to weather conditions, whereas **month** and **year** might be capturing seasonal and yearly trends. **Holiday** indicates whether a day is a holiday or not, which has the least importance among the features shown, suggesting it might have a minimal direct impact on sales in this model.

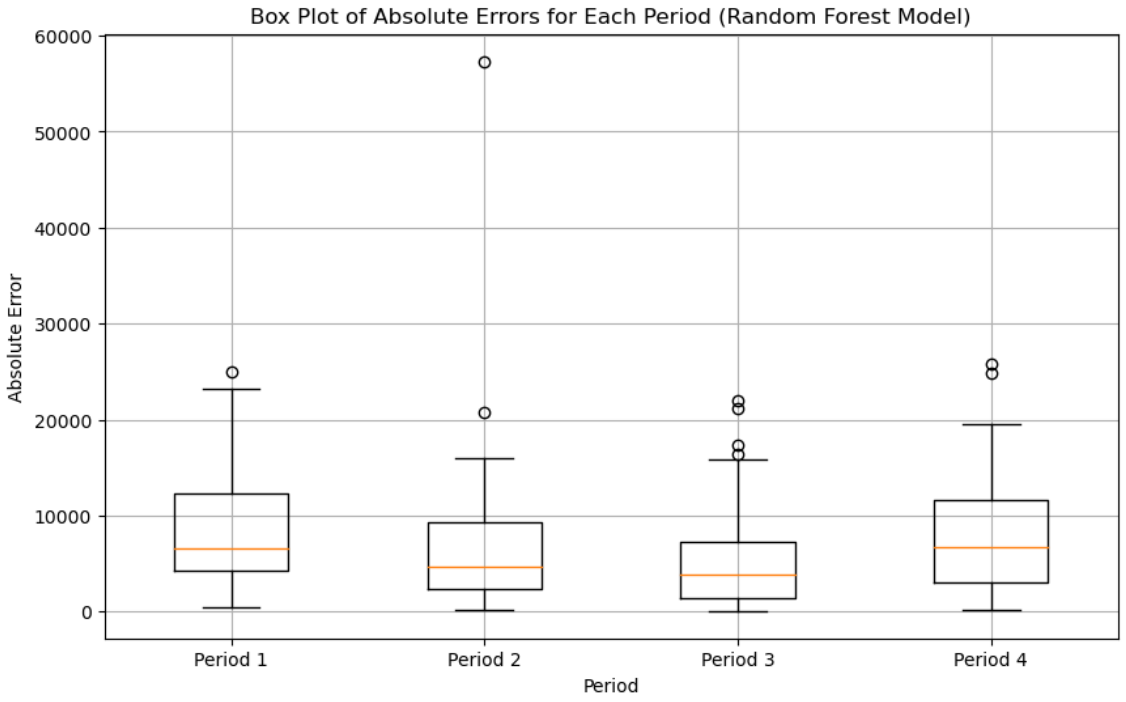
In all, the plot suggests that moving average sales with different time windows and temperature are strong predictors of future daily sales in this model, while holidays and specific weather conditions like snow depth are less significant.

- MAPE on 4 time periods

The MAPE becomes lower as time passes.

	Period 1	Period 2	Period 3	Period 4
MAPE	0.5647	0.2909	0.2496	0.1833

- Errors of prediction on 4 time periods



This box plot visualizes the distribution of absolute errors of sales predictions made by a Random Forest model across four 4 test time periods. The absolute error is the absolute difference between the predicted sales and the actual sales values.

- **Period 1**: The median error is also higher than in Periods 2 and 3, but similar to Period 4.
- **Period 2**: Displays the widest range of errors, but still maintains a relatively lower median error, similar to Period 1. There are a few outliers, showing that there were some predictions that were far off from the actual values.
- **Period 3**: Shows the smallest range of absolute errors, indicating more consistent and accurate predictions in this period. The median error has decreased compared to Periods 1 and 2.
- **Period 4**: The median error is also higher than in Periods 2 and 3, but similar to Period 1.

Overall, the box plot suggests that the Random Forest model's prediction accuracy doesn't varied a lot across different periods, with 4 periods having similar performance. The outliers in the second period highlight instances where the model's predictions were not as close to the actual sales as most other predictions, and we can observe this point on the plot of model's forecasts V.S. actual test data as well.

Part 9: Future Steps

- **Removing outliers**: From our visual analyses, we've identified several outliers. The negative values indicate product cancellations. While their presence in daily aggregates reflects a balance in sales, they may skew forecasting models, leading to less reliable predictions. To enhance model accuracy, especially in terms of MAPE, we could potentially remove these outliers to provide more consistent data for future predictions.
- **Including more features**: We added additional holiday and weather features for our model to improve the prediction. However, many other features are related to the prediction of future

sales as well. For example, promotional activities, economic indicators, and customer reviews and ratings are all good features to add.

- **Implement deep learning models**: For our future steps, we can implement deep learning models to enhance the accuracy and performance of our sales forecasts, as deep learning has proven highly effective at capturing complex patterns and relationships within large datasets. By utilizing architectures such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks that are adept at processing sequential data, we aim to model the temporal dynamics of sales more precisely. Moreover, we can further explore Transformer models, which have recently set new benchmarks in various domains and achieved state-of-art.