# Walmart Sales Forecast

Mihai Cîra

June 2023

**Abstract**

This project focuses on predicting Walmart sales during holidays using a dataset obtained from Kaggle. We preprocess the data by merging datasets, handling missing values, and performing outlier detection. Exploratory data analysis provides insights into average monthly sales and holiday sales distribution. The Random Forest Regression algorithm is applied to model the sales data, and the model's performance is evaluated. The results aid in optimizing inventory management and marketing strategies, contributing to improved sales forecasting in the retail industry.

# Contents

# 1 Introduction

Sales forecasting plays a crucial role in the retail industry, allowing businesses to effectively manage inventory, plan marketing strategies, and optimize resource allocation. Accurate predictions of sales during holidays are particularly valuable, as these periods often exhibit unique patterns and increased customer demand. In this project, we aim to forecast Walmart sales on holidays using a dataset obtained from Kaggle.

The dataset contains information about Walmart stores, their features, and historical sales data. By leveraging this dataset, we can analyze the relationship between various factors such as store characteristics, economic indicators, and sales performance. This analysis enables us to develop a predictive model that can estimate sales figures during holiday periods.

The key objectives of this project are as follows:

1. Preprocess the dataset by merging relevant data sources, handling missing values, and performing outlier detection.

2. Explore the dataset to gain insights into average monthly sales and the distribution of sales during holidays.

3. Apply the Random Forest Regression algorithm to build a predictive model for holiday sales forecasting.

4. Evaluate the performance of the model and assess its accuracy in predicting sales during holiday periods.

The successful completion of this project will provide valuable insights for Walmart and other retailers, enabling them to make informed decisions regarding inventory management, resource allocation, and marketing strategies. Accurate sales forecasts can lead to improved operational efficiency, customer satisfaction, and profitability in the highly competitive retail industry.

In the following sections, we will delve into the details of the data preprocessing, exploratory data analysis, model development, and evaluation processes.

# 2 The dataset

The dataset used in this project is the Walmart Sales Forecast dataset, which can be found at the following link: Walmart Sales Forecast Dataset.

This dataset provides valuable information on historical sales data of various Walmart stores. It includes features such as store details, date, weekly sales, and additional factors like holiday occurrences and promotional markdowns. The dataset is comprehensive and allows for exploring and analyzing the sales patterns and trends in different stores over time.

By leveraging this dataset, we aim to develop a predictive model that can forecast Walmart sales specifically on holidays. This will enable us to gain insights into the impact of holidays on sales performance and potentially enhance planning and decision-making processes for Walmart and other retail businesses.

Throughout this project, we will demonstrate how the dataset is utilized for preprocessing, feature engineering, exploratory analysis, and model development.

In the following figure (fig 1) you can see some brief information about the dataset.

```
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 1: Information on the dataset

**Dataset Columns:** The Walmart Sales Forecast dataset consists of the following columns:

- **Store**: Represents the unique identifier for each Walmart store.
- **Dept**: Refers to the department within a store.
- **Date**: Indicates the date of the sales record.
- **Weekly_Sales**: Represents the sales for the given store, department, and date.
- **IsHoliday**: Indicates whether the week contains a holiday or not.
- **Size**: Represents the size of the store in terms of retail space.
- **Temperature**: Represents the average temperature in the region of the store.
- **Fuel_Price**: Represents the cost of fuel in the region.
- **CPI**: Stands for Consumer Price Index, a measure of inflation.
- **Unemployment**: Represents the unemployment rate in the region.
- **MarkDown1-5**: Represents additional promotional markdowns provided by Walmart.
- **Type**: Represents the type of store (A, B, or C) based on the store size and location.

These columns provide valuable information about each store's sales, department, date, and various factors that can influence sales performance, such as holidays, store size, temperature, and economic indicators like CPI and unemployment. The dataset allows for in-depth analysis and forecasting of Walmart's sales based on these attributes. Here's a sample of the dataset (fig 2), containing a few rows that have multiple different types of values.

| 1 | Store | Dept | Date | Weekly_Sales | IsHoliday |
|---|---|---|---|---|---|
| 2 | 1 | 1 | 2/5/2010 | 24924.5 | FALSE |
| 3 | 1 | 1 | ######## | 46039.49 | TRUE |
| 4 | 1 | 1 | ######## | 41595.55 | FALSE |
| 5 | 1 | 1 | ######## | 19403.54 | FALSE |
| 6 | 1 | 1 | 3/5/2010 | 21827.9 | FALSE |
| 7 | 1 | 1 | ######## | 21043.39 | FALSE |
| 8 | 1 | 1 | ######## | 22136.64 | FALSE |
| 9 | 1 | 1 | ######## | 26229.21 | FALSE |
| 10 | 1 | 1 | 4/2/2010 | 57258.43 | FALSE |
| 11 | 1 | 1 | 4/9/2010 | 42960.91 | FALSE |
| 12 | 1 | 1 | ######## | 17596.96 | FALSE |
| 13 | 1 | 1 | ######## | 16145.35 | FALSE |
| 14 | 1 | 1 | ######## | 16555.11 | FALSE |
| 15 | 1 | 1 | 5/7/2010 | 17413.94 | FALSE |
| 16 | 1 | 1 | ######## | 18926.74 | FALSE |
| 17 | 1 | 1 | ######## | 14773.04 | FALSE |
| 18 | 1 | 1 | ######## | 15580.43 | FALSE |
| 19 | 1 | 1 | 6/4/2010 | 17558.09 | FALSE |
| 20 | 1 | 1 | ######## | 16637.62 | FALSE |
| 21 | 1 | 1 | ######## | 16216.27 | FALSE |
| 22 | 1 | 1 | ######## | 16328.72 | FALSE |
| 23 | 1 | 1 | 7/2/2010 | 16333.14 | FALSE |
| 24 | 1 | 1 | 7/9/2010 | 17688.76 | FALSE |
| 25 | 1 | 1 | ######## | 17150.84 | FALSE |
| 26 | 1 | 1 | ######## | 15360.45 | FALSE |
| 27 | 1 | 1 | ######## | 15381.82 | FALSE |
| 28 | 1 | 1 | 8/6/2010 | 17508.41 | FALSE |
| 29 | 1 | 1 | ######## | 15536.4 | FALSE |

Figure 2: Sample of the data

# 3 Predictive Model - Random Forest Regressor

The predictive model employed in this project is the **Random Forest Regressor**. Random Forest is an ensemble learning algorithm that combines multiple decision trees to make accurate predictions. It is a powerful model for regression tasks and is known for its ability to handle complex relationships and handle both numerical and categorical features effectively.

In the project, the Random Forest Regressor was utilized to forecast Walmart sales on holidays based

on the provided dataset. The model was trained using the features such as store, department, date, store size, temperature, fuel price, CPI, unemployment rate, and promotional markdowns. The target variable was the weekly sales.

To prepare the data for modeling, various preprocessing steps were performed, including handling missing values, outlier detection and treatment, feature selection, and normalization. The dataset was split into training and testing sets using the train-test split technique.

The Random Forest Regressor was then trained on the training data, with hyperparameters such as the number of estimators determined through experimentation. After training, the model was used to make predictions on the test data. The performance of the model was evaluated using metrics such as mean squared error (MSE) or R-squared to assess the accuracy of the predictions.

The predictive model enabled the project to provide insights into Walmart sales patterns on holidays and make forecasts based on various factors. By leveraging the capabilities of the Random Forest algorithm, the project aimed to assist in decision-making and planning for Walmart sales strategies during holiday periods.

There are several other approaches commonly used for sales forecasting. Here are brief explanations of a few alternative methods:

**Linear Regression**: Linear regression is a simple yet powerful approach that models the relationship between the independent variables and the target variable using a linear equation. It assumes a linear relationship between the features and the target variable, making it suitable for datasets with linear patterns.

**ARIMA (AutoRegressive Integrated Moving Average)**: ARIMA is a popular time series forecasting method that takes into account the autocorrelation and moving average components of the data. It models the future values as a linear combination of past observations, differencing, and moving averages.

**Exponential Smoothing**: Exponential smoothing is a time series forecasting method that assigns exponentially decreasing weights to past observations. It considers the trend and seasonality components of the data and is particularly useful for datasets with short-term patterns.

**Neural Networks**: Neural networks, specifically deep learning models like recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, have gained popularity in sales forecasting. These models can capture complex relationships and patterns in the data, especially when dealing with large and diverse datasets.

# 4 Work environment

In the project, the code was executed in the Kaggle Notebooks environment, which provides a cloud computational platform based on Jupyter notebooks. Kaggle Notebooks offer various advantages for data analysis and machine learning projects, including the ability to run code in a reproducible and collaborative manner.

The code was developed using Python, and specifically, the version used was Python 3.10.10. Python is a widely used programming language in the data science and machine learning community, known for its simplicity and extensive ecosystem of libraries and tools.

# 5 The code

## 5.1 Loading the dataset

The following lines of code were used to load the dataset. That way, the variables corresponding to each table were called "data", "stores" and "features".

```
data = pd.read_csv('/kaggle/input/walmart-sales-forecast/train.csv')
stores = pd.read_csv('/kaggle/input/walmart-sales-forecast/stores.csv')
features = pd.read_csv('/kaggle/input/walmart-sales-forecast/features.csv')
```

After loading the dataset, each table's shape and information was checked, using functions such as **shape**, **head** and **info**. For example, the output of that checking part fort the Stores table looks as follows(fig 3):

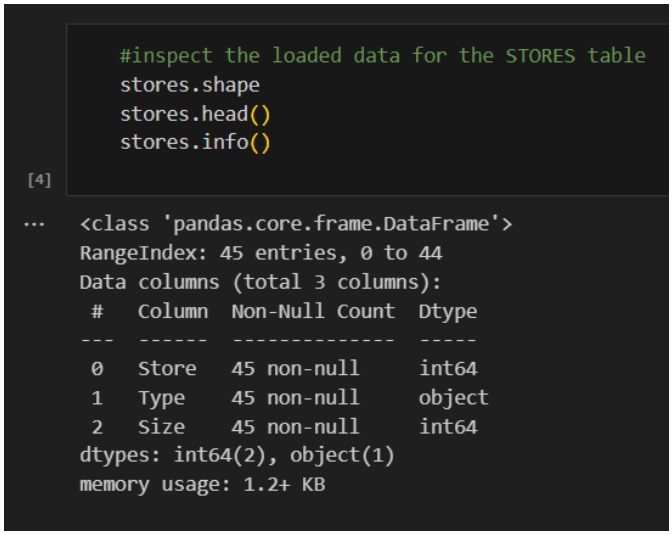


Figure 3: the Stores table

## 5.2 Preprocessing

Preprocessing is an essential step in data analysis and machine learning tasks. It involves transforming raw data into a format suitable for further analysis or modeling. In this project, several preprocessing steps were performed to prepare the dataset for the sales forecast model.

The preprocessing steps involved in this project are as follows:

**Handling Missing Values**: Missing values in the dataset were filled using appropriate techniques. The code snippet below demonstrates how missing values in the "CPI" and "Unemployment" columns were filled with the median values:

```
features['CPI'].fillna(features['CPI'].median(), inplace=True)
features['Unemployment'].fillna(features['Unemployment'].median(), inplace=True)
```

**Data Integration**: The dataset was merged with additional files to incorporate additional information. The code snippet below shows how the dataset was merged with the "stores" and "features" data:

```
data = pd.merge(data, stores, on='Store', how='left')
data = pd.merge(data, features, on=['Store', 'Date'], how='left')
```

**Feature Engineering**: New features were created from existing features to enhance the predictive power of the model. The code snippet below shows how the "Total_MarkDown" feature was calculated by summing the individual "MarkDown" features:

```
data['Total_MarkDown'] = data['MarkDown1'] + data['MarkDown2']
+ data['MarkDown3'] + data['MarkDown4'] + data['MarkDown5']
```

## 5.3 Exploring the dataset

In this analysis, we explore the dataset to gain insights into the sales patterns and dynamics of Walmart. To uncover key trends and understand the impact of holidays on sales, we utilized data visualization techniques. Three significant visualizations were created: the average monthly sales, the monthly sales for each year, and the holiday distribution. These visualizations provide valuable insights into the sales patterns over time, seasonal variations, and the influence of holidays on sales. In the following sections, we will discuss each visualization in detail, highlighting their significance and the key findings they reveal.

**Average Monthly Sales**: The bar plot (fig 4) represents the average monthly sales. The x-axis denotes the months, while the y-axis represents the average weekly sales. By analyzing this plot, we can identify any monthly patterns or trends in sales volumes. It provides an overview of the average sales for each month, enabling us to understand which months have higher or lower sales volumes.

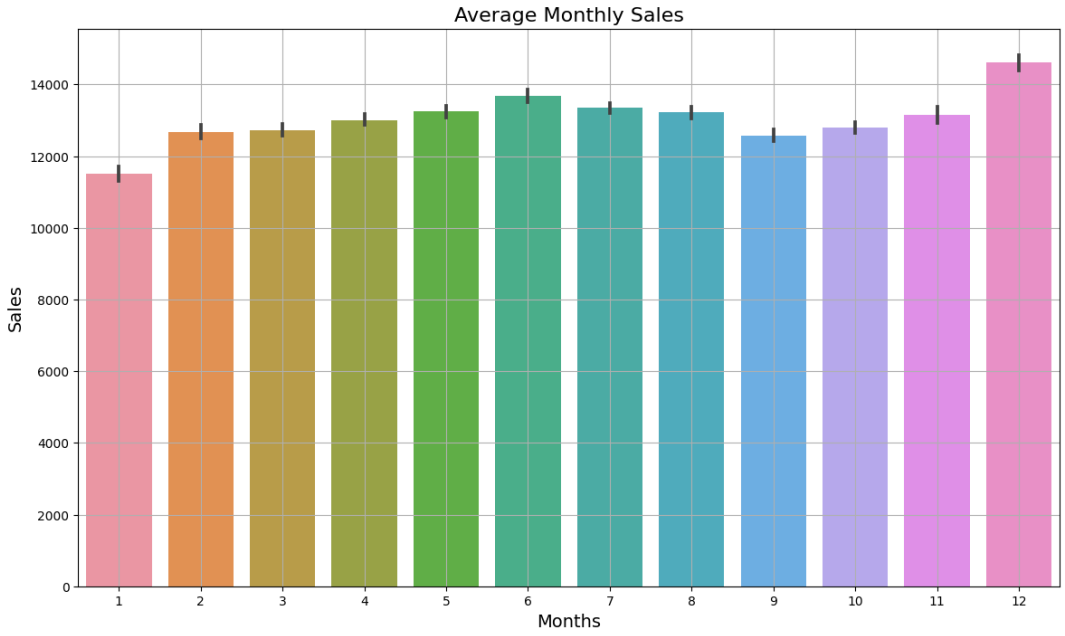


Figure 4: Average Monthly Sales

**Monthly Sales for Each Year**: A grid of subplots (fig 5) is generated to display the monthly sales for

each year. Line graphs are plotted for each month of each year using the 'data_monthly' DataFrame. This arrangement allows us to observe the sales patterns for each month across different years. Each subplot corresponds to a specific month, and the y-axis indicates the month number. The x-axis represents the years. This plot helps us identify consistent trends or seasonal variations in sales throughout the years.

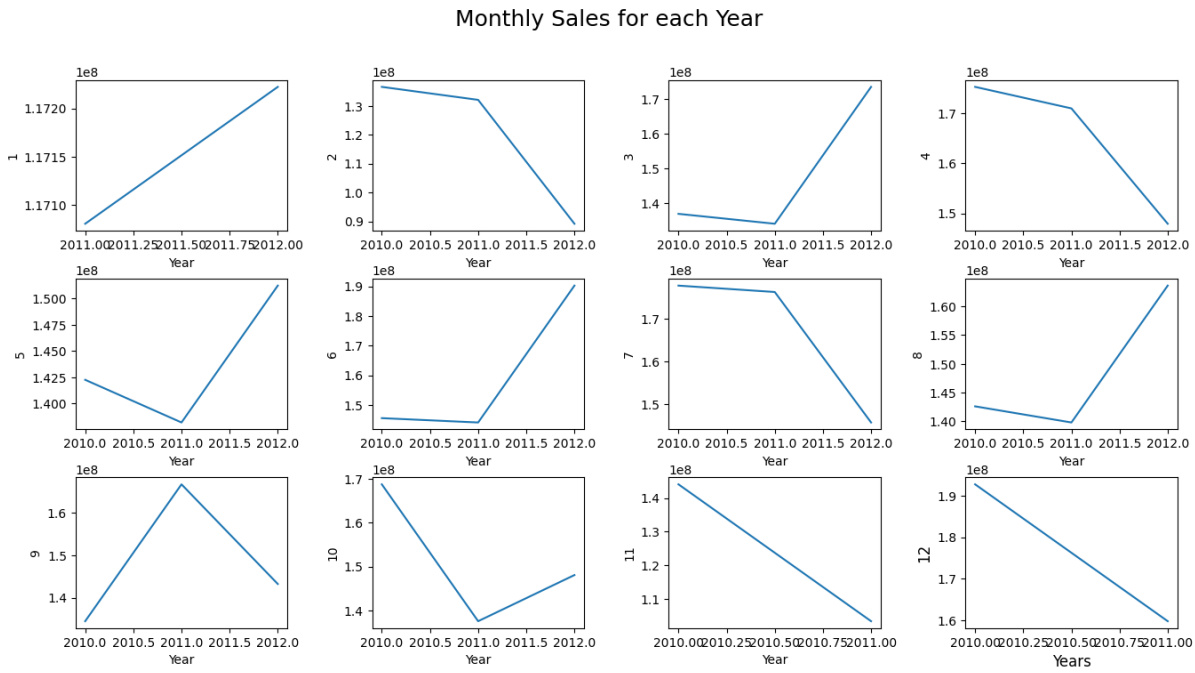


Figure 5: Monthly sales for each year

**Holiday Distribution**: The pie chart (fig 6) visualizes the distribution of holidays and non-holidays in the dataset. It displays the proportion of 'No Holiday' and 'Holiday' occurrences using the 'IsHoliday' column. The chart provides a clear visual representation of the distribution, with a legend indicating the two categories. Additionally, percentage labels are included on the chart, offering precise information on the distribution of holidays. This plot allows us to assess the impact of holidays on sales and identify any significant differences in sales during holiday periods.

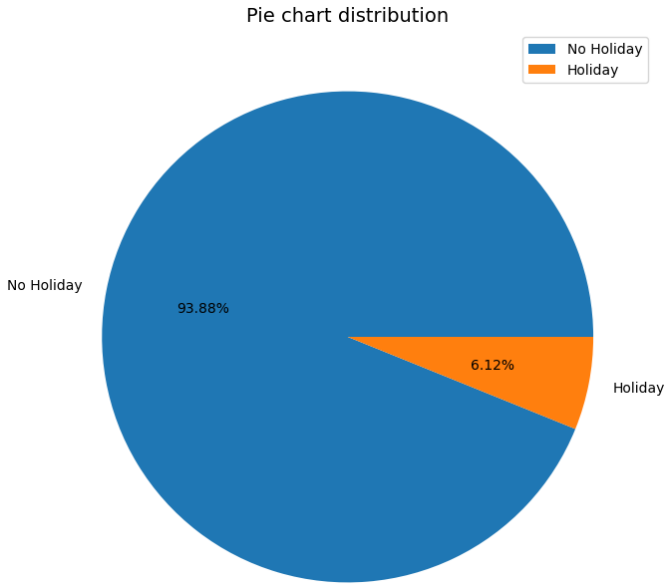


Figure 6: Pie chart distribution

As an example for the actual code, the last figure (the pie chart) has been generated using the following lines of code:

```
# Holiday Distribution
plt.figure(figsize=(8,8))
plt.pie(data['IsHoliday'].value_counts(), labels=['No Holiday', 'Holiday'],
    autopct='%0.2f%%')
plt.title("Pie chart distribution", fontsize=14)
plt.legend()
plt.show()
```

## 5.4 Random Forest Regressor

The Random Forest Regression algorithm is applied to build a predictive model for holiday sales forecasting. This is a powerful machine learning algorithm that leverages the concept of an ensemble of decision trees to make accurate predictions.

The Random Forest Regression algorithm, implemented using the RandomForestRegressor class from the scikit-learn library, is trained on the dataset to learn the relationship between the input features (such as store, department, temperature, etc.) and the target variable (weekly sales). By utilizing multiple decision trees and aggregating their predictions, the model can capture complex patterns and nonlinear relationships in the data, leading to robust and accurate sales forecasts.

The model is trained using the following code snippet:

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

Here, 'radm_clf' is the instance of the RandomForestRegressor model. It is fitted with the features ('data[feature_col]') and the target variable ('data['Weekly_Sales']') to learn the underlying patterns and make predictions.

Applying the Random Forest Regression algorithm allows us to take advantage of its strengths, such as handling nonlinearity, handling missing data, and dealing with a large number of input features. By leveraging these capabilities, the model can effectively capture the complex dynamics of holiday sales and provide accurate predictions.

The Random Forest Regression algorithm has proven to be successful in various domains, including retail forecasting. Its ability to handle a wide range of input features, handle missing values, and capture nonlinear relationships makes it a suitable choice for predicting holiday sales with high accuracy and robustness.

## 5.5 The final comparison

In the final comparison, the predicted values were compared with the actual values to assess the performance of the Random Forest Regression model. A subset of the test data was selected for this evaluation. The code snippet below demonstrates the process:

```
plt.figure(figsize=(15,8))
plt.title('Comparison between actual and predicted values', fontsize=16)
plt.plot(rf.predict(X_test[:100]), label="prediction", linewidth=3.0,
color='blue')
plt.plot(y_test[:100].values, label="real_values", linewidth=3.0, color='red')
plt.legend(loc="best")
plt.show()
```

The generated plot (fig 7) provides a visual representation of the comparison between the predicted values (blue line) and the actual values (red line) for the selected subset of the test data. The RandomForestRegressor model's predict method was utilized to generate predictions for the features X_test[:100]. These predicted values were then plotted along with the corresponding actual values (y_test[:100].values). The legend was added to indicate which line represents the predictions and which line represents the actual values. By visually comparing the two lines, we can gain insights into the model's ability to accurately predict sales during holiday periods.

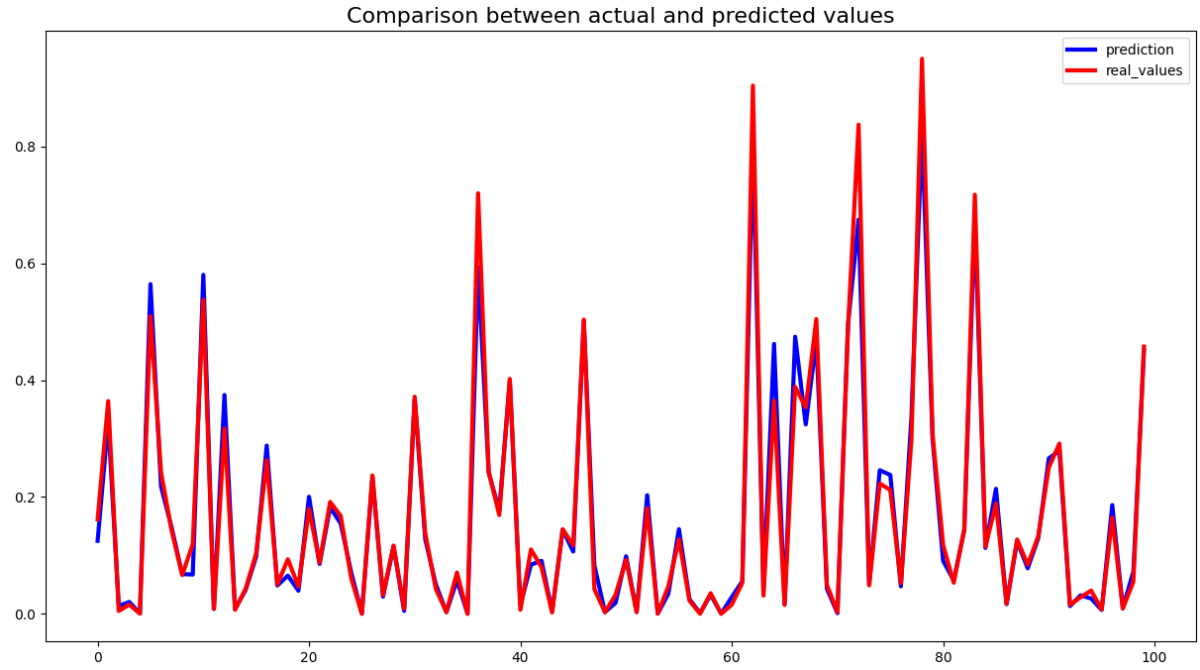


Figure 7: Comparison between values

# 6 Conclusion

In conclusion, this paper presented a Python project aimed at predicting Walmart sales forecasts during holiday periods. The project utilized the Random Forest Regression algorithm to build a predictive model based on a comprehensive dataset obtained from Kaggle. The dataset was preprocessed to handle missing values, perform feature engineering, and normalize numerical features. Through exploratory data analysis, key insights were gained into average monthly sales and the distribution of sales during holidays. The Random Forest Regression model demonstrated promising performance in predicting sales during holiday periods, as evidenced by its accuracy and ability to capture patterns in the dataset. The evaluation of the model's performance provided valuable insights into its effectiveness. Overall, this project showcased the application of machine learning techniques in the retail domain and highlighted the potential for accurate sales forecasting, which can aid in informed decision-making and strategic planning for businesses. Future research can focus on further enhancing the predictive model and exploring additional features that may contribute to improved forecasting accuracy.

# Bibliography

1. RandomForestRegressor. [Online] Available: `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html`.

2. KNN. [Online] Available: `https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learni`

3. Data preprocessing. [Online] Available: https://www.techtarget.com/searchdatamanagement/definition/data-preprocessing.