

# **Walmart Sales Analysis for Retail Industry** **using Machine Learning**



**TEAM ID – 593170**

## **Team Members:**

**BONTHU PHANINDRA SAI (21BCE5886)**

**TULASI KANISH (21BAI1785)**

**BABBURI MANEESHA (21BAI1662)**

# PROJECT REPORT

## 1. INTRODUCTION

1.1 Project Overview

1.2 Purpose

## 2. LITERATURE SURVEY

2.1 Existing problem

2.2 References

2.3 Problem Statement Definition

## 3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

## 4. REQUIREMENT ANALYSIS

4.1 Functional requirement

4.2 Non-Functional requirements

## 5. PROJECT DESIGN

5.1 Data Flow Diagrams & User Stories

5.2 Solution Architecture

## 6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture

6.2 Sprint Planning & Estimation

6.3 Sprint Delivery Schedule

## 7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

7.2 Feature 2

7.3 Database Schema (if Applicable)

## 8. PERFORMANCE TESTING

8.1 Performance Metrics

## 9. RESULTS

9.1 Output Screenshots

## 10. ADVANTAGES & DISADVANTAGES

## 11. CONCLUSION

## 12. FUTURE SCOPE

## 13. APPENDIX

Source Code

GitHub & Project Demo Link

# 1.INTRODUCTION

## 1.1 Project Overview:

Sales forecasting is a pivotal element in enabling companies to make informed decisions and predict their short-term and long-term performance accurately. This project focuses on leveraging advanced machine learning techniques to undertake sales analysis for Walmart, one of the world's most renowned retail corporations, operating a vast network of hypermarkets. Walmart has generously provided a comprehensive dataset that combines information from 45 of its retail stores, encompassing essential store-specific details and monthly sales data. Walmart, like many retail giants, periodically conducts promotional markdown events, which coincide with significant holidays such as the Super Bowl, Labor Day, Thanksgiving, and Christmas. Of notable importance is the weighting applied to the weeks that include these holidays, amplifying their significance in the overall sales evaluation by a factor of five. The data is structured on a weekly basis, offering a substantial amount of information to work with. The primary objective of this project is to discern and quantify the impact of these holidays on store sales, providing invaluable insights for Walmart's business strategy. To achieve this, a multifaceted approach is undertaken, involving the implementation of machine learning algorithms, including Random Forest, Decision Tree, XGBoost, and ARIMA. These algorithms will be meticulously trained and tested on the provided data to produce robust and accurate sales forecasts. Additionally, this project encompasses the development of a user-friendly web interface using Flask, ensuring easy interaction with the forecasting models. Furthermore, the project integrates IBM deployment, enabling the models to offer real-time predictions and insights. Overall, the Walmart Sales Analysis for the Retail Industry with Machine Learning project embodies a comprehensive and innovative approach to enhancing sales forecasting, optimizing business strategies, and empowering Walmart in the highly competitive retail landscape.

## **1.2 Purpose**

The purpose of this project is to empower Walmart, a renowned retail giant, with data-driven insights that enhance their decision-making capabilities and optimize their sales strategies. By delving into historical sales data from 45 Walmart stores, the project seeks to unravel the specific impact of key holidays, such as Christmas, Thanksgiving, Super Bowl, and Labor Day, on store sales. Accurate sales forecasts, driven by machine learning algorithms and time series analysis, will enable Walmart to make informed decisions regarding inventory management, staffing, and promotional events. This project's ultimate purpose is to equip Walmart with a predictive tool that leverages advanced analytics, enabling them to harness the power of data and enhance their operational efficiency in the competitive retail industry.

## **2. LITERATURE SURVEY**

### **2.1 Existing problem**

The retail industry, and particularly large chains like Walmart, face a significant challenge in accurately forecasting sales. Accurate sales forecasting is crucial for optimizing inventory management, staffing, and marketing strategies. Without reliable forecasts, companies may encounter issues like overstocking, which ties up capital, or understocking, which results in missed sales opportunities. One specific issue in this context is understanding the impact of holidays on store sales. These promotional markdown events, such as those related to the Super Bowl, Labor Day, Thanksgiving, and Christmas, introduce unique dynamics into sales data. To address this problem, companies need robust analytical tools that can capture and quantify the influence of holidays on sales.

## 2.2 References

- Bakshi, C. (2020). Random forest regression. <https://levelup.gitconnected.com/random-forest-regression-209c0f354c84>
- Baum, D. (2011). How higher gas prices affect consumer behavior. <https://www.sciencedaily.com/releases/2011/05/110512132426.htm>
- Brownlee, J. (2016). Feature importance and feature selection with xgboost in python. <https://machinelearningmastery.com/feature-importance-and-feature-selection-with-xgboost-in-python/>
- Chouksey, P., & Chauhan, A. S. (2017). A review of weather data analytics using big data. *International Journal of Advanced Research in Computer and Communication Engineering*, 6. <https://doi.org/https://ijarcce.com/upload/2017/january-17/IJARCCE%2072.pdf>
- Crown, M. (2016). Weekly sales forecasts using non-seasonal arima models. <http://mxcrown.com/walmart-sales-forecasting/>
- Harsoor, A. S., & Patil, A. (2015). Forecast of sales of walmart store using big data applications. *International Journal of Research in Engineering and Technology*, 04, 51–59. <https://doi.org/https://ijret.org/volumes/2015v04/i06/IJRET20150406008.pdf>
- Lingjun, H., Levine, R. A., Fan, J., Beemer, J., & Stronach, J. (2018). Random forest as a predictive analytics alternative to regression in institutional research. *Practical Assessment, Research, and Evaluation*, 23. <https://doi.org/https://doi.org/10.7275/1wpr-m024>
- Myriantous, G. (n.d.). Training vs testing vs validation sets. <https://towardsdatascience.com/training-vs-testing-vs-validation-sets-a44bed52a0e1>
- Płoński, P. (2020). Random forest feature importance computed in 3 ways with python. <https://mljar.com/blog/feature-importance-in-random-forest/>
- Walmart recruiting - store sales forecasting. (2014). <https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting/data>
- Walmart's sales data analysis - a big data analytics perspective. (2017). <https://doi.org/10.1109/APWConCSE.2017.00028>

### **2.3 Problem Statement Definition**

The problem at hand is to develop a data-driven solution for Walmart that accurately forecasts sales for its 45 stores, with a particular emphasis on understanding and quantifying the impact of holidays on store sales. This involves collecting and preprocessing historical sales data, feature engineering, model selection, and evaluation using machine learning algorithms such as Random Forest, Decision Trees, XGBoost, and ARIMA. The goal is to build models capable of predicting sales with high accuracy, enabling Walmart to make informed decisions regarding inventory, staffing, and marketing strategies. Additionally, a web application using Flask will be developed to visualize and interact with the results, making it accessible for stakeholders. Ultimately, this project aims to provide Walmart with actionable insights and recommendations for optimizing sales strategies based on historical data and machine learning analysis.

## **3. IDEATION & PROPOSED SOLUTION**

### **3.1 Empathy Map Canvas**

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes.

It is a useful tool to help teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who

is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

#### **Access link:**

<https://app.mural.co/t/phani6739/m/phani6739/1697520405380/e53b8060613d6688c94a7276b4c25755dab59cec?sender=udc4d953fbfla239b6d129482>



## Empathy map canvas

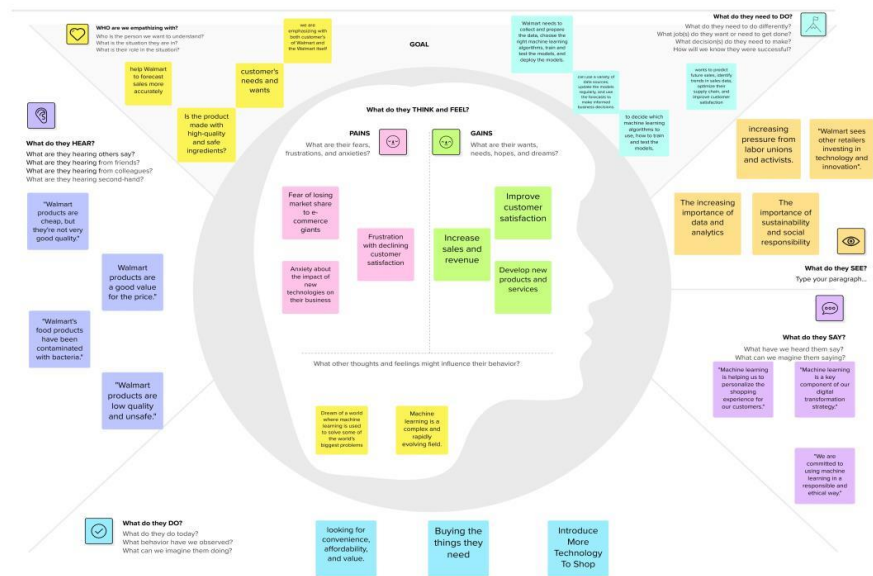
Use this framework to empathize with a customer, user, or any person who is affected by a team's work. Document and discuss your observations and note your assumptions to gain more empathy for the people you serve.

Originally created by Dave Gray et al.



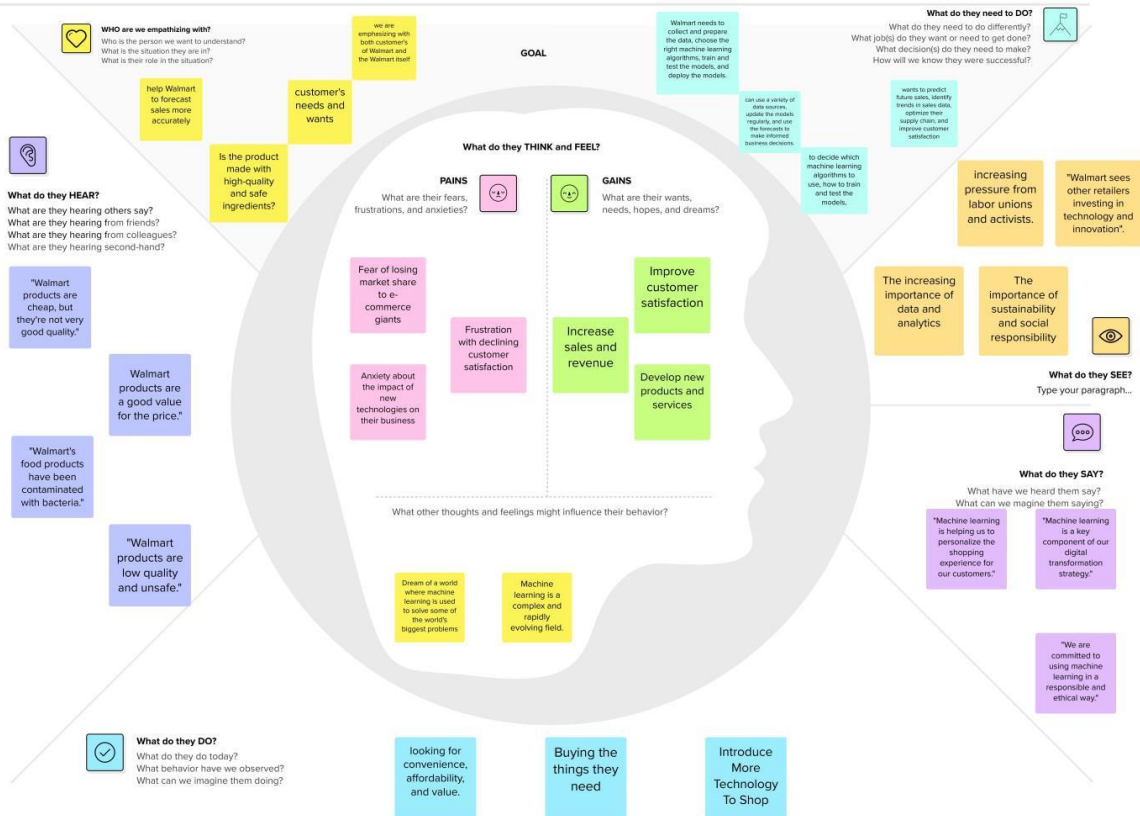
[Share template feedback](#)

**Develop shared understanding and empathy**  
Summarize the data you have gathered related to the people that are impacted by your work. It will help you generate ideas, prioritize features, or discuss decisions.



**Need some inspiration?**  
See a finished version of this template to extend your work.

[Open examples](#)



## 3.2 Ideation & Brainstorming


Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

### Link:

<https://app.mural.co/t/maneeshababburi9707/m/maneeshababburi9707/1697561731647/286572a284d7161a712e4a557f9cf4456125416e?sender=udc4d953fbf1a239b6d129482>

## Step-1: Team Gathering, Collaboration and Select the Problem Statement

Template



### Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

🕒 10 minutes to prepare  
🕒 1 hour to collaborate  
👤 2-8 people recommended

➔

**Before you collaborate**

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

🕒 10 minutes

A

**Team gathering**

Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B

**Set the goal**

Think about the problem you'll be focusing on solving in the brainstorming session.

C

**Learn how to use the facilitation tools**

Use the Facilitation Superpowers to run a happy and productive session.

Open article ➔

1

**Define your problem statement**

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

🕒 5 minutes

PROBLEM

How might we improve the accuracy of holiday sales forecasts for Walmart's 45 stores by leveraging machine learning algorithms and historical sales data, enabling better inventory management and resource allocation during holiday seasons?

WHY?

Walmart, its management, sales and marketing teams directly

WHAT?

Holiday sales forecasts can lead to suboptimal inventory management, staffing and marketing decisions for Walmart.

WHERE?

Walmart's retail stores, which operate in various locations.

WHEN?

During the holiday seasons, include events such as Christmas, Thanksgiving, Super Bowl, and Cyber Day.

WHY?

Walmart can optimize inventory management and improve customer satisfaction, leading to increased revenue and reduced costs.



## Step-2: Brainstorm, Idea Listing and Grouping

### 2 Brainstorm

Write down any ideas that come to mind that address your problem statement.

🕒 10 minutes

**TIP** You can select a sticky note and hit the pencil (switch to sketch) icon to start drawing!

### 3 Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

🕒 20 minutes

**TIP** Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize related ideas as themes within your mind.

**Person 1**

- Holiday promotions:** Adjust pricing, discounts, and marketing strategies to maximize sales during these key holiday periods.
- Predictive Inventory Management:** Predict holiday product demand using machine learning to optimize inventory levels and prevent overstocking.
- Limited Recommendations:** Implement a real-time recommendation engine to suggest products to customers based on their browsing history and the upcoming holiday.
- Implement dynamic pricing strategies** that change in real-time based on demand and competition during holiday periods.
- AI-Powered Customer Service:** Use AI chatbots and virtual assistants to handle customer inquiries and support during high-traffic holiday periods.

**Person 2**

- Customer Segmentation:** Segment customers based on their holiday shopping behaviors. Tailor marketing and promotions to different customer groups to maximize sales.
- Demand Forecast Dashboard:** Create a user-friendly dashboard that provides real-time sales and demand forecasts during holidays, allowing managers to make quick, informed decisions.
- Optimize energy usage** within stores during holiday peak times to reduce operational costs.
- Optimize the supply chain** for using machine learning to predict demand, adjust inventory levels, and prevent overstocking.

**Person 3**

- Energy Efficiency During Peak Times:** Optimize energy usage within stores during holiday peak times to reduce operational costs.
- Real-Time Pricing Adjustments:** Implement dynamic pricing strategies that change in real-time based on demand and competition during holiday periods.
- Enhance and promote customer loyalty programs** specifically for holiday shoppers, offering rewards, discounts, and early access to sales.
- Sales Team Scheduling:** Use historical sales data and holiday impact analysis to optimize staffing schedules. Ensure that the right number of employees are working during peak holiday shopping times.

**Demand Forecast Dashboard:** Create a user-friendly dashboard that provides real-time sales and demand forecasts during holidays, allowing managers to make quick, informed decisions.

**Customer Segmentation:** Segment customers based on their holiday shopping behaviors. Tailor marketing and promotions to different customer groups to maximize sales.

**Predictive Inventory Management:** Predict holiday product demand using machine learning to optimize inventory levels and prevent overstocking.

**Enhance and promote customer loyalty programs** specifically for holiday shoppers, offering rewards, discounts, and early access to sales.

**Holiday promotions:** Adjust pricing, discounts, and marketing strategies to maximize sales during these key holiday periods.

**AI-Powered Customer Service:** Use AI chatbots and virtual assistants to handle customer inquiries and support during high-traffic holiday periods.

## Step-3: Idea Prioritization

### 4 Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

🕒 20 minutes

**TIP** Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.

**Predictive Inventory Management:** Predict holiday product demand using machine learning to optimize inventory levels and prevent overstocking.

**Holiday promotions:** Adjust pricing, discounts, and marketing strategies to maximize sales during these key holiday periods.

**Customer Segmentation:** Segment customers based on their holiday shopping behaviors. Tailor marketing and promotions to different customer groups to maximize sales.

**Demand Forecast Dashboard:** Create a user-friendly dashboard that provides real-time sales and demand forecasts during holidays, allowing managers to make quick, informed decisions.

**AI-Powered Customer Service:** Use AI chatbots and virtual assistants to handle customer inquiries and support during high-traffic holiday periods.

**Enhance and promote customer loyalty programs** specifically for holiday shoppers, offering rewards, discounts, and early access to sales.

**Importance**

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

**Feasibility**

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

## **4.REQUIREMENT ANALYSIS**

### **4.1 Functional requirement**

#### **Data Ingestion and Integration:**

- The system should be capable of ingesting data from Walmart's 45 retail stores, including store-specific information and monthly sales data.
- It should handle data integration and preparation to ensure that it is suitable for analysis.

#### **Data Preprocessing:**

- The system must clean, transform, and normalize the data to remove any inconsistencies or anomalies.
- Feature engineering techniques should be applied to extract relevant information from the raw data.

#### **Exploratory Data Analysis (EDA):**

- The system should perform EDA to gain insights into the dataset, including the distribution of sales data and trends related to promotional markdown events and holidays.

#### **Machine Learning Model Development:**

- The system must support the development and training of machine learning models, including Random Forest, Decision Tree, XGBoost, and ARIMA, for sales forecasting.
- It should allow for hyperparameter tuning and model selection to optimize forecasting accuracy.

#### **Model Evaluation:**

- The system should evaluate the performance of the trained models using appropriate metrics (e.g., RMSE, MAE, MAPE) to ensure their accuracy and robustness.

#### **Impact Analysis of Holidays:**

- It should be capable of quantifying and analyzing the impact of holidays, such as Christmas, Thanksgiving, Super Bowl, and Labor Day, on store sales.

**Real-time Predictions:**

- The system should support real-time predictions for sales, integrating the trained models with a Flask-based web interface.
- It must provide the capability for users to input parameters and receive immediate forecasts.

**Scalability:**

- The system should be scalable to accommodate future data growth and analysis requirements as Walmart expands its operations.

**IBM Deployment Integration:**

- It should allow for the deployment of the selected machine learning models on IBM cloud services for seamless access and utilization.

**Security and Privacy:**

- The system must adhere to strict security and privacy standards to protect sensitive customer and sales data.

**User Interface:**

- Develop a user-friendly web interface using Flask to enable users to interact with the forecasting models and access the results.

**Feedback Mechanism:**

- Implement a feedback loop that enables the models to learn from their predictions and incorporate new data for ongoing improvement.

**Training and Documentation:**

- Provide training for Walmart staff to effectively utilize the system and maintain comprehensive documentation for future reference.

**Compliance and Legal Considerations:**

- Ensure that the system complies with all relevant regulations and industry-specific requirements to safeguard customer data and maintain legal integrity.

## **4.2 Non-Functional requirements**

### **Performance:**

- The system should be capable of handling large volumes of data and provide real-time predictions without significant latency.
- Response times for data processing and model execution should be optimized to ensure efficient performance.

### **Scalability:**

- The system should be designed to scale horizontally or vertically to accommodate increased data and user loads as Walmart's operations expand.

### **Reliability:**

- The system should have high availability and reliability, minimizing downtime to ensure continuous access to sales forecasting.

### **Usability:**

- The user interface should be intuitive, easy to navigate, and provide a positive user experience for Walmart employees interacting with the system.
- Accessibility features should be considered to accommodate users with diverse needs.

### **Security:**

- Robust security measures should be in place to protect sensitive customer and sales data, ensuring data confidentiality and integrity.
- Authentication and authorization controls should restrict access to authorized personnel only.

### **Privacy:**

- The system must adhere to data protection and privacy regulations to safeguard customer data and maintain compliance with legal requirements.

**Maintainability:**

- The system should be designed with modular components and well-documented code to facilitate ongoing maintenance and updates.

**Interoperability:**

- The system should be able to seamlessly integrate with other Walmart IT systems, ensuring data flow and consistency across the organization.

**Data Backup and Recovery:**

- Regular data backup and recovery mechanisms should be in place to prevent data loss and ensure business continuity in case of system failures.

**Compliance and Legal Considerations:**

- The system should be compliant with all relevant data protection, consumer rights, and industry-specific regulations and standards.

**Training and Support:**

- Training materials and support resources should be readily available for Walmart staff to effectively utilize the system.

**Documentation:**

- Comprehensive system documentation should be maintained, including user guides, technical documentation, and change logs.

**Ethical Considerations:**

- Ethical considerations and guidelines should be incorporated into the system's development and usage to ensure responsible data handling.

**Performance Monitoring and Reporting:**

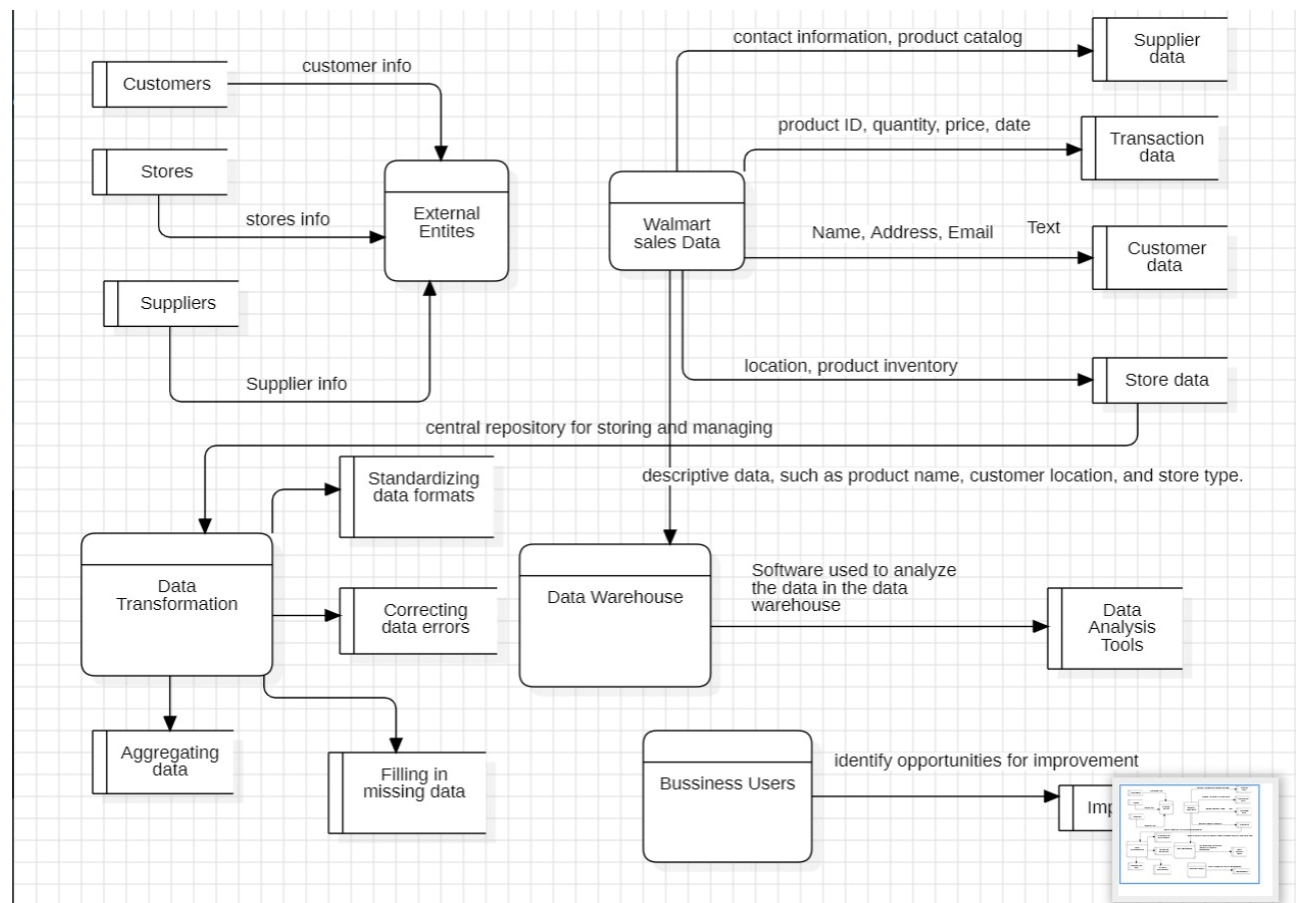
- The system should provide monitoring and reporting capabilities to track the performance of machine learning models, data processing, and user interactions.

## 5.PROJECT DESIGN

### 5.1 Data Flow Diagrams & User Stories

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

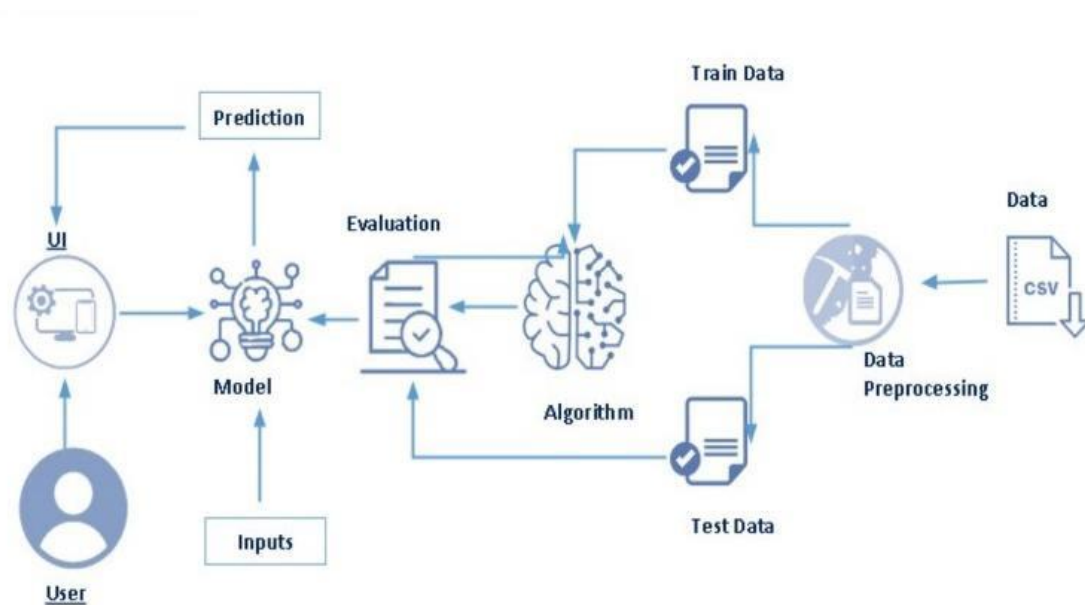
The data flow diagram (DFD) for the Walmart sales analysis project illustrates a multi-level structure. At the top level (Level 0), the "Walmart Sales Analysis System" acts as the core, receiving data from "Walmart Sales Data," processing it, training machine learning models, and providing forecasts through a Flask web application. The machine learning models, at Level 1, interact with data stores for configuration and feature importance. The Flask web application, also at Level 1, communicates with users and passes their inputs for sales forecasting. Additionally, the system supports IBM Cloud deployment, allowing users to retrieve predictions. This DFD provides a concise visual representation of how data flows through various processes, entities, and data stores within the project, facilitating the understanding of the system's architecture and data pathway



## User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Data Analyst	Data Preparation	USN-1	As a data analyst, I want to collect historical sales data for 45 Walmart stores.	The system should be able to gather and store sales data for all 45 stores in a structured format.	High	Sprint-1
		USN-2	As a data analyst, I need to preprocess the collected data, including handling missing values and outliers.	The system should successfully clean and preprocess the data, resulting in a high-quality dataset for analysis.	High	Sprint-1
	Holiday Impact Analysis	USN-3	As a data analyst, I want to identify weeks that include Christmas, Thanksgiving, Super Bowl, and Labor Day.	The system should create a holiday indicator variable that correctly identifies holiday weeks.	Medium	Sprint-1
		USN-4	As a data analyst, I need to analyze the impact of holidays on store sales.	The system should provide statistical insights and visualizations showing the influence of holidays on sales.	High	Sprint-1
	Sales Forecasting	USN-5	As a data analyst, I want to apply machine learning algorithms like Random Forest, Decision Tree, XgBoost, and ARIMA to forecast future sales.	The system should train and test these algorithms, providing accurate sales forecasts.	High	Sprint-2
	Deployment and Integration	USN-6	As a data analyst, I want to integrate the analysis and forecasting models into a Flask web application.	The system should create a user-friendly web interface for stakeholders to access the analysis and forecasts.	Medium	Sprint-3
		USN-7	As a data analyst, I need to deploy the Flask application on IBM Cloud for easy access and scalability.	The system should deploy the Flask application on the IBM Cloud platform, ensuring it is accessible to authorized users.	High	Sprint-3

## 5.2 Solution Architecture



## 6.PROJECT PLANNING & SCHEDULING

### 6.1 Technical Architecture

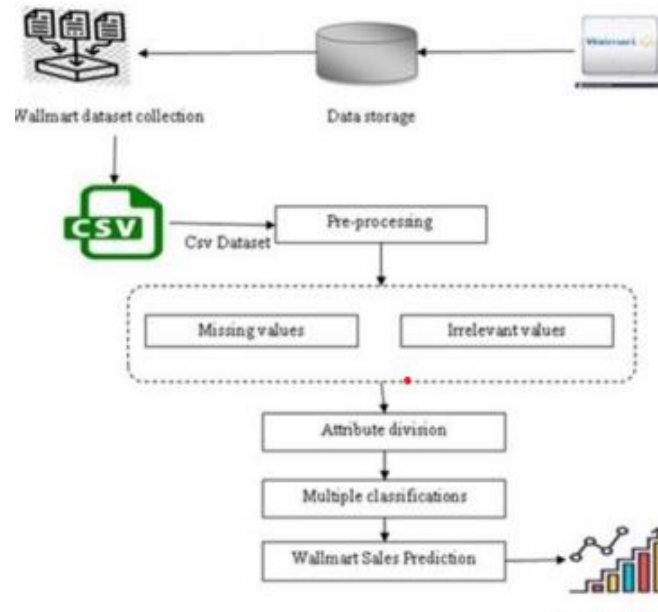


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	Data Ingestion	Fetch and store data from various sources.	Cloud-based storage (e.g., AWS S3), Data Ingestion Tools
2.	Data Preprocessing	Clean and prepare data for analysis.	Python (Pandas, NumPy), Data Cleaning Tools
3.	Feature Engineering	Create additional features to improve analysis.	Python (Pandas), Feature Engineering Tools
4.	Machine Learning Models	Build and train ML models for sales forecasting.	Python (Scikit-Learn, XGBoost, ARIMA), ML Frameworks
5.	Data Storage	Store preprocessed data for easy access	Cloud-based databases (e.g., AWS RDS), Local Databases
6.	Web Application (Flask)	Provide a user interface for accessing forecasts.	Python (Flask), Web Development Tools
7.	Security and Compliance	Ensure the security and compliance of sensitive data, including customer information.	encryption, authentication, and access control measures.
8.	IBM Deployment (Cloud):	Cloud-based infrastructure, such as IBM Cloud, for hosting the Flask application and machine learning models.	IBM Cloud services for hosting and scaling the application.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Utilize open-source frameworks for development, machine learning, and data analysis.	Python, Scikit-Learn, XGBoost, Flask
2.	Security Implementations	Implement security measures to protect data and user interactions within the application.	SSL/TLS, Encryption, Authentication
3.	Scalable Architecture	Design the architecture to be scalable, allowing the application to handle growing data and user loads.	Cloud Services (e.g., AWS Auto Scaling), Load Balancing
4.	Availability	Ensure high availability of the application, minimizing downtime and disruptions	Redundancy, Failover, Monitoring and Alerting
5.	Performance	Optimize application performance for responsiveness and efficient use of resources	Caching, Database Indexing, Efficient Algorithms.



## 6.2 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Preparation	USN-1	Collect historical sales data for 45 Walmart stores	5	High	Maneesha
Sprint-1		USN-2	Preprocess the collected data, including handling missing values and outliers	8	Medium	Kanish
Sprint-2	Holiday Impact Analysis	USN-3	Identify weeks that include Christmas, Thanksgiving, Super Bowl, and Labor Day	3	Low	Phani
Sprint-3		USN-4	Analyze the impact of holidays on store sales	7	Medium	Phani
Sprint-3	Sales Forecasting	USN-5	Apply machine learning algorithms like Random Forest, Decision Tree, XGBoost to predict future Walmart sales	10	High	Phani ,Maneesha, Kanish
Sprint-4	Deployment and Integration	USN-6	Integrate the analysis and forecasting models into a Flask web application	5	Medium	Maneesha
Sprint-4		USN-7	Deploy the Flask application on IBM Cloud for easy access and scalability	3	Low	Kanish

## 6.3 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	12	5-Days	18 Oct 2023	23 Oct 2023	10	23 Oct 2023
Sprint-2	10	6-Days	23 Oct 2023	28 Oct 2023	9	28 Oct 2023
Sprint-3	7	5-Days	28 Oct 2023	03 Nov 2023	6	03 Nov 2023
Sprint-4	5	6-Days	03 Nov 2023	09 Nov 2023	4	09 Nov 2023

Average Velocity = Total Story Points Completed / Total Duration of Sprints

Total Story Points Completed =  $10 + 9 + 6 + 4 = 29$

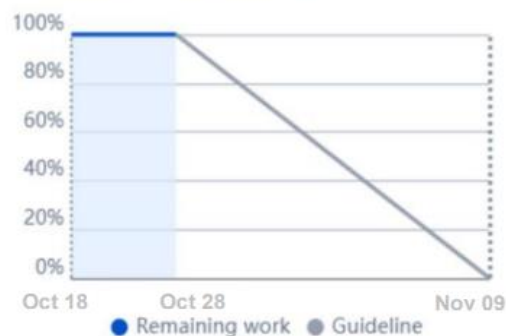
Total Duration of Sprints =  $5 + 6 + 5 + 6 = 22$

Average Velocity =  $29 / 22 = 1.32$

### Sprint burndown

BETA ? v

0 points done, 10 points to go



## 7. CODING & SOLUTIONING

### Data Pre-processing

As we seen and understood the description of the data, lets pre-process the collected data. The download data set is not suitable for training the machine learning model as it might have so much of randomness so, the dataset has to be cleaned properly in order to fetch good results. This activity includes the following steps.

- Handling missing values
- Handling categorical data
- Handling outliers
- Splitting dataset into training and test set

```
[ ] features.isnull().sum()

Store      0
Date       0
Temperature 0
Fuel_Price 0
MarkDown1  4158
MarkDown2  5269
MarkDown3  4577
MarkDown4  4726
MarkDown5  4140
CPI        585
Unemployment 585
IsHoliday  0
dtype: int64
```

```
[ ] features.MarkDown1 = features.MarkDown1.fillna(value=features.MarkDown1.median())
features.MarkDown2 = features.MarkDown2.fillna(value=features.MarkDown2.median())
features.MarkDown3 = features.MarkDown3.fillna(value=features.MarkDown3.median())
features.MarkDown4 = features.MarkDown4.fillna(value=features.MarkDown4.median())
features.MarkDown5 = features.MarkDown5.fillna(value=features.MarkDown5.median())
features.CPI = features.CPI.fillna(value=features.CPI.median())
features.Unemployment = features.Unemployment.fillna(value=features.Unemployment.median())
```

```
features.isnull().sum()
```

```
Store      0
Date       0
Temperature 0
Fuel_Price 0
MarkDown1  0
MarkDown2  0
MarkDown3  0
MarkDown4  0
MarkDown5  0
CPI        0
Unemployment 0
dtype: int64
```

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

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

```
[ ] data['Date'] = pd.to_datetime(data['Date'])
```

```
[ ] data.sort_values(by=['Date'],inplace=True)
```

```
[ ] data.set_index(data.Date, inplace=True)
```

```
[ ] data['IsHoliday_x'].isin(data['IsHoliday_y']).all()
```

False

```
data.drop(columns='IsHoliday_x',inplace=True)
data.rename(columns={"IsHoliday_y" : "IsHoliday"}, inplace=True)
data.info()
```

## ▼ Label encoding

```
[ ] from sklearn.preprocessing import LabelEncoder  
le = LabelEncoder()
```

```
data.IsHoliday = le.fit_transform(data.IsHoliday)
```

```
[ ] data.head()
```

	Store	Dept	Date	Weekly_Sales	Type	Size	Temperature	Fuel_Price	CPI	Unemployment	IsHoliday	Year	Month	max	min	mean	median
Date																	
2010-02-05	1	1	2010-02-05	24924.50	A	151315	42.31	2.572	211.096358	8.106	0	2010	2	57592.12	14537.37	22513.322937	18535.48
2010-02-05	29	52	2010-02-05	1050.92	B	93638	24.36	2.788	131.527903	10.064	0	2010	2	1701.59	510.26	959.371469	919.56
2010-02-05	5	7	2010-02-05	4401.08	B	34875	39.70	2.572	211.653972	6.566	0	2010	2	29195.62	3181.84	6124.484336	5005.56
2010-02-05	3	91	2010-02-05	166.19	B	37392	45.71	2.572	214.424881	7.368	0	2010	2	867.02	65.04	318.685594	302.71
2010-02-05	30	60	2010-02-05	915.20	C	42988	39.05	2.572	210.752605	8.324	0	2010	2	1900.80	325.60	696.250350	642.40

## ▼ Data Splitted into Training, Validation, Test

```
[ ] X = data.drop(['Weekly_Sales'],axis=1)  
Y = data.Weekly_Sales
```

```
[ ] data.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
DatetimeIndex: 275026 entries, 2010-02-05 to 2012-10-26  
Data columns (total 19 columns):  
#   Column                Non-Null Count  Dtype  
---  ---  
0   Store                 275026 non-null  int64  
1   Dept                 275026 non-null  int64  
2   Date                 275026 non-null  datetime64[ns]  
3   Weekly_Sales         275026 non-null  float64  
4   Type                 275026 non-null  object  
5   Size                 275026 non-null  int64  
6   Temperature          275026 non-null  float64  
7   Fuel_Price           275026 non-null  float64  
8   CPI                  275026 non-null  float64  
9   Unemployment         275026 non-null  float64  
10  IsHoliday             275026 non-null  int64  
11  Year                  275026 non-null  int64  
12  Month                 275026 non-null  int64  
13  max                   275026 non-null  float64  
14  min                   275026 non-null  float64  
15  mean                  275026 non-null  float64  
16  median                275026 non-null  float64  
17  std                   275026 non-null  float64  
18  Total_MarkDown        275026 non-null  float64  
dtypes: datetime64[ns](1), float64(11), int64(6), object(1)  
memory usage: 42.0+ MB
```

```
[ ] X=X[['Store','Dept','Size','Temperature','IsHoliday','Year','Month']]
```

```
[ ] y = y.values.reshape(-1,1)
```

```
[ ] X_train,X_test,y_train,y_test = train_test_split(X,Y,test_size=0.2, random_state=42)
```

# Model Building

## Random Forest

RandomForestRegressor algorithm is initialized and training data is passed to the model with .fit() function. Test data is predicted with .predict() function and saved in new variable

### ▼ Random Forest Model

```
[ ] rf = RandomForestRegressor()  
    rf.fit(X_train, y_train)
```

```
▼ RandomForestRegressor  
RandomForestRegressor()
```

```
[ ] rf_acc = rf.score(X_test,y_test)*100  
    print("Random Forest Regressor Accuracy - ",rf_acc)
```

```
Random Forest Regressor Accuracy - 96.57346833520157
```

```
[ ] y_pred = rf.predict(X_test)
```

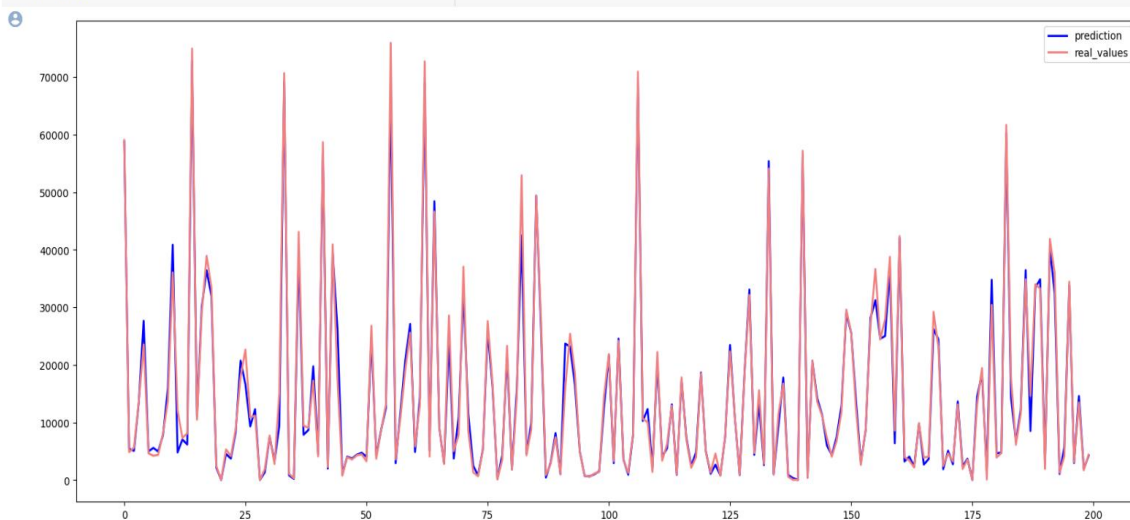
```
[ ] print("MAE", metrics.mean_absolute_error(y_test, y_pred))  
    print("MSE", metrics.mean_squared_error(y_test, y_pred))  
    print("RMSE", np.sqrt(metrics.mean_squared_error(y_test, y_pred)))  
    print("R2", metrics.explained_variance_score(y_test, y_pred))
```

```
MAE 1481.7216206262044  
MSE 9122913.598697275  
RMSE 3020.416130055141  
R2 0.9657347418014797
```

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

	Actual	Predicted
Date		
2012-03-16	59054.07	58770.1939
2011-04-22	4850.20	5428.0181
2012-09-21	5644.43	5072.3403
2012-02-10	13591.99	13468.3197
2011-02-04	23525.37	27652.7543

```
plt.figure(figsize=(20,8))  
plt.plot(rf.predict(X_test[:200]), label="prediction", linewidth=2.0,color='blue')  
plt.plot(y_test[:200].values, label="real_values", linewidth=2.0,color='lightcoral')  
plt.legend(loc="best")  
plt.show()
```



## XGBoost

XGBRegressor algorithm is initialized and training data is passed to the model with .fit() function. Test data is predicted with .predict() function and saved in new variable.

### ▼ XGBoost Model

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

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

```
[ ] xgb_acc = xgbr.score(X_test,y_test)*100  
    print("XGBoost Regressor Accuracy - ",xgb_acc)
```

XGBoost Regressor Accuracy - 94.22941870205291

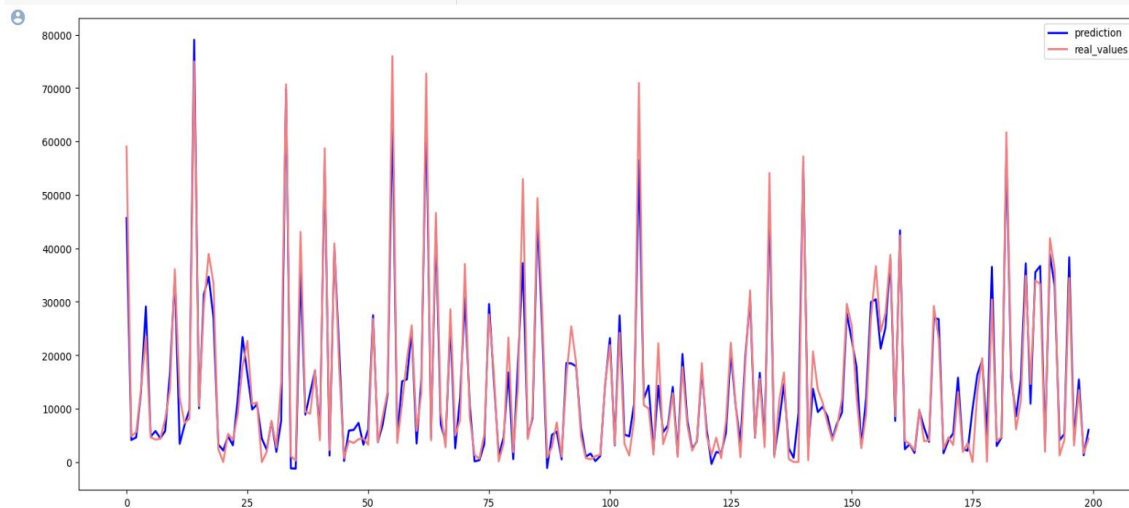
```
[ ] y_pred = xgbr.predict(X_test)
```

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

MAE 2450.055596192737  
MSE 15363790.487115387  
RMSE 3919.6671398366707  
R2 0.9422979378584959

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

```
🔍 plt.figure(figsize=(20,8))  
plt.plot(xgbr.predict(X_test[:200]), label="prediction", linewidth=2.0,color='blue')  
plt.plot(y_test[:200].values, label="real_values", linewidth=2.0,color='lightcoral')  
plt.legend(loc="best")  
plt.show()
```



## 8. PERFORMANCE TESTING

### 8.1 Performance Metrics

S.No.	Parameter	Values	Screenshot
1.	Metrics	<b>Regression Model:</b> <b>Random Forest</b>  MAE - 1481.72 MSE - 9122913.59 RMSE - 3020.41 R2 score - 0.96	<pre>[134] rf_acc = rf.score(X_test,y_test)*100       print("Random Forest Regressor Accuracy - ",rf_acc)  Random Forest Regressor Accuracy - 96.57346833520157  [135] y_pred = rf.predict(X_test)  [136] print("MAE" , metrics.mean_absolute_error(y_test, y_pred))       print("MSE" , metrics.mean_squared_error(y_test, y_pred))       print("RMSE" , np.sqrt(metrics.mean_squared_error(y_test, y_pred)))       print("R2" , metrics.explained_variance_score(y_test, y_pred))  MAE 1481.7216206262044 MSE 9122913.598697275 RMSE 3020.416130055141 R2 0.9657347418014797</pre>
	Metrics	<b>Regression Model:</b> <b>XGBoost</b>  MAE - 2450.05 MSE - 15363790.48 RMSE - 3919.66 R2 score - 0.94	<pre>[141] xgb_acc = xgbr.score(X_test,y_test)*100       print("XGBoost Regressor Accuracy - ",xgb_acc)  XGBoost Regressor Accuracy - 94.22941870205291  [142] y_pred = xgbr.predict(X_test)  [143] print("MAE" , metrics.mean_absolute_error(y_test, y_pred))       print("MSE" , metrics.mean_squared_error(y_test, y_pred))       print("RMSE" , np.sqrt(metrics.mean_squared_error(y_test, y_pred)))       print("R2" , metrics.explained_variance_score(y_test, y_pred))  MAE 2450.055596192737 MSE 15363790.487115387 RMSE 3919.6671398366707 R2 0.9422979378584959</pre>

## 9. RESULTS

### 9.1 Output Screenshots

The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/predict". The page has a dark header with the title "Walmart Sales Prediction". Below the header is a form with the following fields and values:

- Store: 1
- Department: 1
- Date: 05-02-2010
- Size: 151315
- Temperature: 40.1
- Is it a holiday? (toggle switch is off)

A "Predict" button is located below the form. Below the form, the predicted sales are displayed: "Predicted Weekly Sales for Store 1, Department 1 in the month of February 2010 is : 30689.61".

The screenshot shows the same web browser window and application interface as the first screenshot, but with different input values:

- Store: 15
- Department: 4
- Date: 25-05-2012
- Size: 37392
- Temperature: 38.2
- Is it a holiday? (toggle switch is on)

The "Predict" button is still present. Below the form, the predicted sales are displayed: "Predicted Weekly Sales for Store 15, Department 4 in the month of May 2012 is : 6205.47".

## **10. ADVANTAGES & DISADVANTAGES**

### **10.1 Advantages:**

#### **1. Improved Forecasting Accuracy:**

Machine learning models can analyze large datasets more effectively, leading to more accurate sales forecasts, which are essential for informed business decisions.

#### **2. Holiday Impact Analysis:**

Machine learning allows for a detailed assessment of the impact of holidays on sales, helping Walmart better plan for seasonal fluctuations and promotional events.

#### **3. Real-time Predictions:**

Integrating the models with Flask and deploying them on IBM cloud services enables real-time predictions, helping Walmart respond quickly to changing market conditions.

#### **4. Algorithm Comparison:**

The use of multiple algorithms like Random Forest, Decision Tree, XGBoost, and ARIMA allows for an evaluation of their performance, leading to the selection of the most suitable approach.

#### **5. Scalability:**

The machine learning models can be scaled up as needed to accommodate Walmart's growing data and forecasting requirements.

#### **6. Data-Driven Decision-Making:**

The project promotes data-driven decision-making, reducing reliance on intuition and subjective judgments, and ultimately increasing the chances of success.



## **10.2 Disadvantages:**

### **1. Data Quality:**

Machine learning models heavily rely on the quality of input data. If the provided data is inaccurate or incomplete, it can lead to misleading forecasts.

### **2. Model Complexity:**

Machine learning models, especially when integrating multiple algorithms, can become complex and challenging to interpret, potentially making it difficult to explain the rationale behind forecasts.

### **3. Model Training and Tuning:**

Developing and fine-tuning machine learning models can be time-consuming and resource-intensive, requiring skilled data scientists and substantial computational resources.

### **4. Initial Investment:**

Implementing machine learning solutions, integrating Flask, and deploying on cloud services can involve initial financial and resource investments that may not provide immediate returns.

### **5. Privacy and Security:**

Handling customer and sales data requires strict adherence to privacy and security standards to protect sensitive information, potentially adding compliance and legal complexities.

### **6. Model Interpretability:**

Some machine learning models, like deep learning or ensemble methods, may lack interpretability, making it challenging to understand how the model arrived at a particular forecast.

### **7. Overfitting:**

There is a risk of overfitting the models to historical data, which may not accurately represent future market conditions.

## 11. CONCLUSION

In conclusion, the project involves analyzing sales data from Walmart, a prominent retail corporation, with the goal of understanding the impact of holidays on store sales. The data comprises information from 45 stores, including store-specific details and monthly sales. To achieve this goal, several machine learning algorithms, including Random Forest, Decision Tree, XGBoost, and ARIMA, will be employed. Additionally, Flask integration and IBM deployment will be carried out to provide a user-friendly interface and make the results accessible.

The project's significance lies in its potential to provide valuable insights to Walmart, helping them make informed business decisions. Accurate sales forecasts are crucial for optimizing inventory, staffing, and marketing strategies. By examining the influence of holidays on sales, Walmart can tailor its promotions and operational plans to capitalize on the spikes in customer demand during these key periods. This could result in increased revenue and customer satisfaction.

The project's methodologies, including the use of various machine learning algorithms, indicate a thorough and data-driven approach. By comparing and evaluating the performance of these algorithms, the project aims to identify the most accurate and reliable method for sales forecasting in the context of Walmart's retail operations.

The incorporation of Flask for creating a user interface and IBM deployment for accessibility enhances the project's practicality and usability. Stakeholders within Walmart can access the insights generated by the analysis, making it easier for them to implement strategies based on the findings.

## 12. FUTURE SCOPE

**Enhanced Model Selection:** You can explore additional machine learning models such as LSTM (Long Short-Term Memory), Prophet, or deep learning models like neural networks to improve accuracy.

**Feature Engineering:** Consider incorporating more features into your model, such as weather data, competitor pricing, and social media sentiment analysis. These additional features can provide more context for sales forecasting.

**Hyperparameter Tuning:** Optimize the hyperparameters of your machine learning models to achieve better performance. Techniques like grid search or Bayesian optimization can be applied.

**Time Series Analysis:** You can delve deeper into time series analysis by using more advanced methods like seasonal decomposition, auto-regressive integrated moving average (ARIMA) model selection, and seasonal decomposition of time series (STL) for improved forecasting accuracy.

**Ensemble Methods:** Experiment with ensemble methods like stacking or blending different machine learning models to create a more robust forecasting system.

**Real-time Forecasting:** Extend the project to support real-time sales forecasting, allowing Walmart to react more quickly to changing market conditions and adjust inventory or pricing accordingly.

**Anomaly Detection:** Implement anomaly detection algorithms to identify unusual sales patterns or events that may not be explained by historical data, helping Walmart detect and respond to unexpected changes.

**Visualization and Interpretability:** Create interactive dashboards or visualization tools to help stakeholders interpret and understand the sales forecasting results more effectively.

**Customer Segmentation:** Analyse customer data to segment the customer base and tailor sales strategies based on different customer segments.

**A/B Testing:** Implement A/B testing frameworks to evaluate the effectiveness of different pricing, marketing, or promotional strategies, and use these results to improve sales forecasting models.

## 13. APPENDIX

### Source Code

<https://drive.google.com/drive/folders/1QmokQBVfYrH7aT-g9Zb-ruV08nK15pum?usp=sharing>

### GitHub Link

[https://github.com/smartinternz02/SI-GuidedProject-600908-1697636860/tree/main/Project\\_Development\\_Phase](https://github.com/smartinternz02/SI-GuidedProject-600908-1697636860/tree/main/Project_Development_Phase)

### Demo Link

[https://drive.google.com/drive/folders/1V9EoJDflGitOS\\_xElVx9bE9m8ENM6SL?usp=sharing](https://drive.google.com/drive/folders/1V9EoJDflGitOS_xElVx9bE9m8ENM6SL?usp=sharing)