Walmart Sales Prediction using Machine Learning

Akhil Katam, Rushikesh Harikishan Kankar, Shwetha Lokesh

***Abstract* - Every day, superstores like Walmart transact an immeasurable amount of goods and money. The managers' most important decision is how to sustain a balance between inventory as well as consumer demand assuming their high transaction rates. As a result, for stores to amplify their revenues, correct sales forecasting for various products turns out to be crucial. Most present-day sales predictions are based especially on extending the statistical fashion. The earlier reports on predicting market sales necessitate a lot more specifics, such as customer as well as product analysis. To foresee product sales using exclusively past sales data, the department store needs a supplementary straightforward model. We can now produce forecasts like that with superior accuracy thanks to lately developed machine learning techniques. For this project, we used the data from Kaggle's competition – M5 Forecasting: Uncertainty. Using the sales data, price data, and calendar data, we will utilize different machine learning models and algorithms such as Random Forest, LightGbm, and XGBoost to estimate the upcoming Walmart sales for a period of 28 days.**

I. INTRODUCTION

A. Background

By using historical sales data, sales forecasting is the practice of predicting future sales, either short- or long-term. For example, spending on new approaches to promote income for their merchandise that may account for a small number of sales in the upcoming days through deals/discounts, etc., foreseeing sales is critical for businesses. Therefore, it becomes crucial for established businesses to precisely predict future sales.

In this project, we will predict daily sales for the following 28 days using hierarchical sales data from Walmart which is one of the largest companies in the world by revenue. This project's goal is to predict product unit sales with great accuracy for Walmart in the United States. The daily sales of the products are predicted using three separate machine-learning models for the next 28 days. Weighted Root Mean Square Scaled Error (WRMSSE) is used to score the predictions made by the model. The predictions of these models, with a good score of WRMSSE, can help the business analyst to better plan for various business-level functions, such as product fulfillment, inventory distribution, distribution management, and storage options.

B. Problem and Importance

The issue is that in a business model like Walmart, inaccurate sales forecasts would cost the company income and business opportunities. The retail stores of Walmart in that location would consequently prepare less storage of the masks for the inventory distribution step of the management process, for instance, if the Walmart analyst team is predicting a flu season and stocks up masks and the model predicts that only a small number of masks are required. However, buyer demand would be more than usual if the estimate turns out to be inaccurate. The cost of shipping, allocating storage for the item, and the lost opportunity for sales, as a result, lower the revenue for those retail businesses Competitors who have the same target market will profit from this. Therefore, having precise machine learning models is crucial when making predictions and deriving business insights. A solution for this issue needs to be developed to eliminate the problem.

C. *Existing Literature*

Many researchers and industry professionals have recently examined sales forecast models. Previously, arithmetical techniques have been engaged to forecast future sales. For sales forecasting, a broad range of arithmetical approaches has been employed, including linear simulations, non-linear models, biased averages, moving averages, and others.

Deep learning is now being used in sales forecasting because of its brilliant results. The deep learning model can extract meaningful data features and produce better prediction outcomes than the conventional statistic approach.

Despite deep learning increasing prediction accuracy, the results of their predictions are challenging to comprehend. Deep neural networks, for instance, replicate intricate non-linear processes by layering numerous networks. The features it retrieves are the variables in each of the network's levels. However, because of how complex these characteristics are for humans to comprehend, there is a limited amount of usable information that can be obtained first from the model performance by the real employees of businesses.

Since they can combine the advantages of many models to create a new forecasting method, multiple hybrid forecasting techniques have been created in past few years. Convolution neural networks as well as fuzzy models are merged in the works of Sadaeit al. in 2019 to forecast short-term time series. Khandelval et al., in 2015, forecast time series using hybrid ARIMA as well as ANN models. Many of them are thought to be more effective than deep learning models and simple statistical models.

D. *Data Collection and Machine Learning System*

The dataset for this implementation was shared by Walmart. This data was monitored and recorded by Walmart for 6 years. This included the number of everyday sales for several products. This data was shared with the public on Kaggle as part of the ‘M5 Forecasting - Uncertainty’ Competition in 2020. [1]

The time series data, which has everyday data for 6 years is split into train and test data. Out of the 1914 days of data, 1886 days is used for training, and 28 days of data is used for testing. 28 days is chosen as Walmart requires us to predict the sales for that number of days, as part of the M5 Competition. This data will be processed and prepared for the machine learning regressor models before training them. The data will be experimented with using multiple models like Random Forest, XGBoost, AdaBoost, and LightGbm.

II. IMPORTANT DEFINITIONS AND PROBLEM STATEMENT

A. *Important Definitions*

1. **Time Series Data**: Time series data is a collection of data that is recorded at regular intervals of time. When plotted on a graph, one axis would always be time. In this case, the data refers to the number of products sold on a daily basis, in a few Walmart locations.
2. **Lag Features**: Lag features are values from earlier timesteps that are deemed beneficial since they were developed under the presumption that the past can affect or contain some sort of intrinsic knowledge about the future.

3. **Rolling Features**: The primary drive of developing as well as employing rolling frame statistics in a period sequence dataset is to compute statistics on the costs from a provided data illustration by establishing a collection that contains the sample as well as some determined number of samples since the sample was used.
4. **Regression Model**: A vital clue in the field of machine learning is something called 'regression analysis.' The structure is learned by exploiting both input sorts as well as output tags, which is 'supervised learning'. Determining how a single attribute influences another, assists in creating a connection between the attributes.

B. Problem Statement

The fifth iteration of the M-competitions was held from March 3rd to June 30th, 2020, over a period of 4 months. The data are given by Walmart and include 30490 hierarchical time series with 1941 days of history, as well as 3049 goods from 3 categories and 7 departments spread over 10 shops in 3 different states.

The M5 competition's objective is to forecast all items 28 days in advance at various aggregation levels (state, store/category, store, department, state/category, state/department, store/department, category, etc.), yielding a total of 42840 times series.

C. The Dataset

The dataset includes group time series that groups together unit deals of numerous products marketed in the United States of America. The dataset consists of the unit sales of 3,049 products, which are broken down into 7 product departments (Foods, Hobbies, and Household), 3 product categories (Hobbies, Foods, and Household), and 3 product groups (Hobbies, Foods, and Household). Ten locations, spread across three States, sell the products (CA, TX, and WI). In this regard, the product-store unit sales at the bottom of the hierarchy can be represented over either product groups or topographical regions, as shown in the following examples:

Level id	Aggregation Level	Number of series
1	Unit sales of all products, aggregated for all stores/states	1
2	Unit sales of all products, aggregated for each State	3
3	Unit sales of all products, aggregated for each store	10
4	Unit sales of all products, aggregated for each category	3
5	Unit sales of all products, aggregated for each department	7
6	Unit sales of all products, aggregated for each State and category	9
7	Unit sales of all products, aggregated for each State and department	21
8	Unit sales of all products, aggregated for each store and category	30
9	Unit sales of all products, aggregated for each store and department	70
10	Unit sales of product *x*, aggregated for all stores/states	3,049
11	Unit sales of product *x*, aggregated for each State	9,147
12	Unit sales of product *x*, aggregated for each store	30,490
Total		**42,840**

Figure 1. The number of M5 series for various aggregations

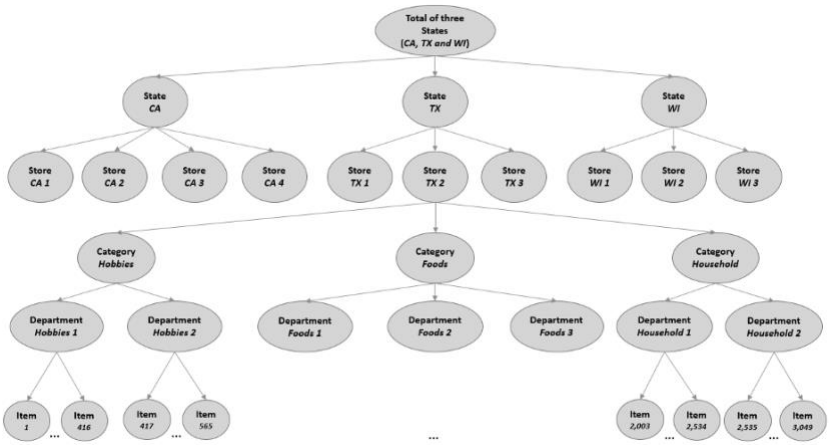

Figure 2. An overview of the dataset organization.

The dataset mainly consists of three files:

File 1: "*calendar.csv*"
Information regarding the days is contained in this file.

- *date*: Each date, is recorded in a year-month-date format.
- *wm_yr_wk*: Week's ID - the date it is part of.
- *weekday*: Day of the week.
- *wday*: ID for the day of the week.
- *month*: Month of year.
- *year*: Year number.
- *event_name_1*: Name of event 1.
- *event_type_1*: Type of event 1.
- *event_name_2*: Name of event 2.
- *event_type_2*: Type of event 2.
- *snap_CA*, *snap_WI*, and *snap_TX*: Snap sales, Boolean value.

File 2: *"sell_prices.csv"*
The file stores the date and the price of the products sold per store.

- *store_id*: The ID of the superstore where the items are sold.
- *item_id*: Item ID.
- *wm_yr_wk*: Week ID.
- *sell_price*: Item price.

File 3: "*sales_train.csv*"
Contains the historical daily unit sales data per product and store.

- *item_id*: Item ID.
- *dept_id*: Department ID.
- *cat_id*: Category ID.
- *store_id*: Store's ID.
- *state_id*: State's ID.
- *d_1, d_2, …, d_i, … d_1941*: Daily sales, starting from 29th January of 2011.

D. Evaluation Metric

For evaluating the performance of the above-mentioned models, we will be using the WRMSSE score, as provided in the same Kaggle competition [1]. It stands for Weighted Root Mean Square Scaled Error. The Formula for this Metric is shown below:

$$\boldsymbol{RMSSE} = \sqrt{\frac{1}{h}\frac{\sum_{t=n+1}^{n+h}\left(Y_t - \hat{Y}_t\right)^2}{\frac{1}{n-1}\sum_{t=2}^{n}(Y_t - Y_{t-1})^2}}$$

Formula 1: Root Mean Square Scaled Error

$$\boldsymbol{WRMSSE} = \sum_{i=1}^{42{,}840} w_i * RMSSE$$

Formula 2: Weighted Root Mean Square Scaled Error

E. Prediction Target and Experimental Results

The sole objective of this project is to predict the figure of sales in a few Walmart stores for the next 28 days while training our machine learning model using the daily sales for the past six years. And as mentioned above, the performance of these models is calculated using the WRMSSE score (Weighted Root Mean Square Scaled Error).

A good WRMSSE score would be under 0.65 and surpassing that was the main ambition behind this project. The dataset, provided by Walmart was used. The data was preprocessed and prepared for the various regressor models. Models experimented with are Random Forest, AdaBoost, XGBoost, and LightGBM. Random Forest and AdaBoost showed poor performance as they took several hours to train the data. Even their accuracy was not as good as XGBoost, that is, a WRMSSE score of 0.667 as opposed to 0.636. However, LightGBM showed the best performance and accuracy. It trained quickly and provided a much better WRMSSE score of 0.615. Which is a huge improvement from the rest. This goes on to show that LightGBM is the best model for this use case.

III. PROPOSED SYSTEM – OVERVIEW AND TECHNICAL DETAILS

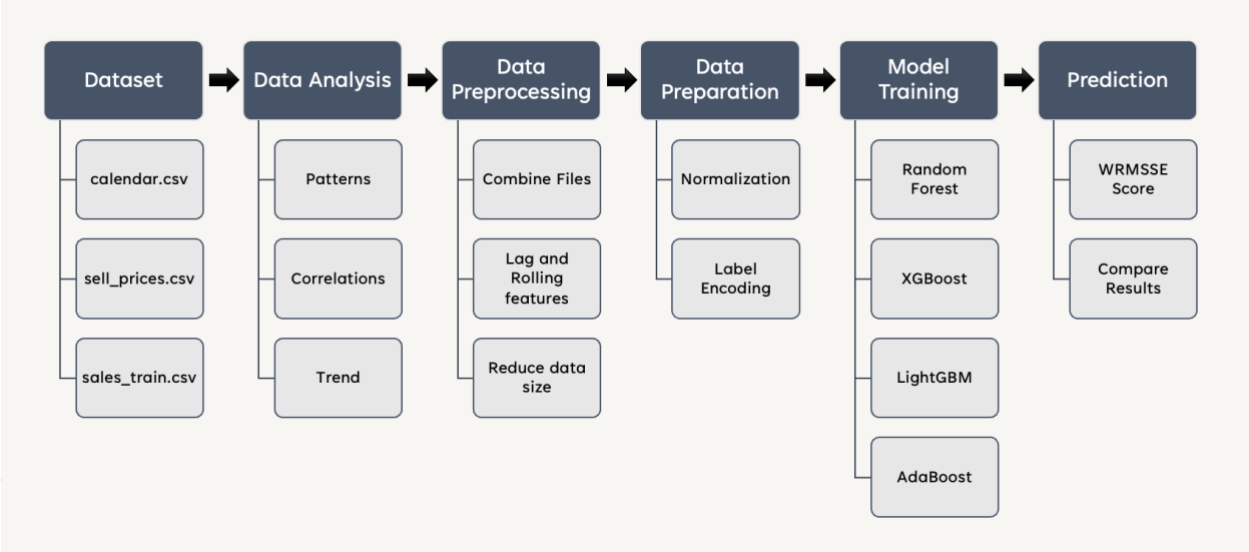

Figure 3: Flow of the proposed system

A. Data Analysis

In this step, we performed all the exploratory data analysis. The main objective of this step was to comprehend the data and find patterns that would lead us to take the right design choices, both for preprocessing and the machine learning models. Once the patterns and few correlations were explored, the data can now be preprocessed and prepared for training the machine learning models.

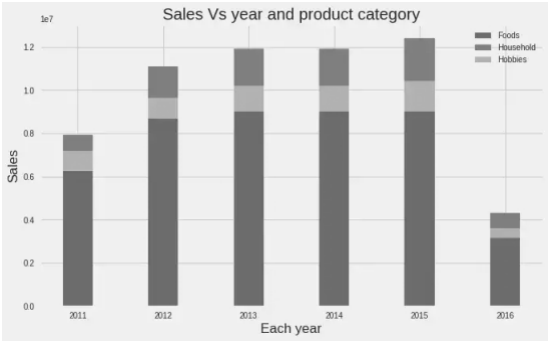

Figure 4: Graph showing food as the most sold product category

Data missing from the years 2011 and 2016. Data is available only from the 29th of January 2011 and six months of data are missing from the year 2016. The data also has more than 17 percent of the items with missing prices.

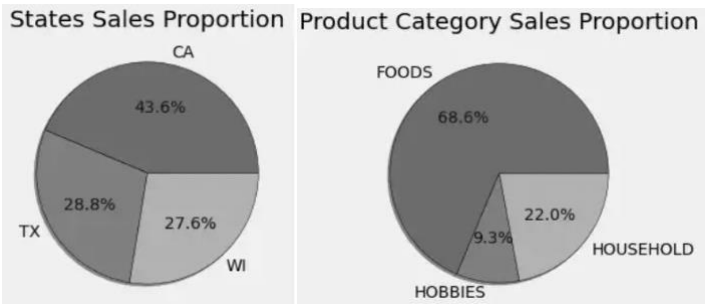

Figure 5: Sales distribution across states and categories

Changes in sales are noticed during various types of events like Christmas, Thanksgiving, or Hurricane. Sales usually increase during such events. More sales are made by customers across states during the first one-half of the month compared to the second one-half of the month. Customers tend to shop more on weekends compared to weekdays.

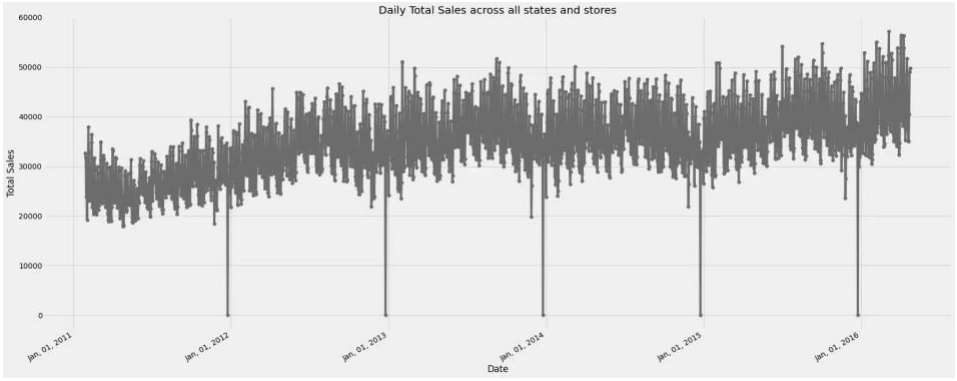

Figure 6: Overall sales trend over the years

From the year 2011 to the year 2016, the overall sales have been trending upward slightly upward. All Walmart stores are shut on New Year's Eve so zero sales on those days. The pattern of sales looks similar every year confirming the idea of seasonality.

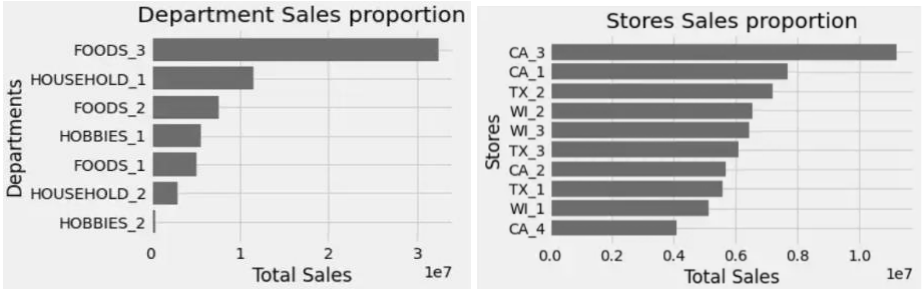

Figure 7: Total sales with respect to departments and store ids

California State accounts for the most sales in the 3 states (California, Texas, and Wisconsin) which is about 43%. The food group has the greatest sales ratio of the 3 categories (Hobbies, Food, and Household) which is about 68.6%.

B. Data Preprocessing

The raw data that was provided by Walmart was far from ready to be fed to machine learning regressor models. The data was split into three files. These three files had to be combined to create a single data frame, as it would be easy to train the models.

Once a single data frame is created, it was time for further processing. The day-wise data was initially stored in columns rather than in rows. That is the reason why the data frame had about two thousand columns. This would not help in training. So, the data frame was modified to have the daily sales recorded in rows rather than columns. So, the number of rows now changed to about fifty-three million.

This massive amount of data consumes forty-two Gigabytes of space. This must be reduced for optimal memory usage. The dataset is then reduced by limiting the space taken by each datatype. By doing the same for every column, the data frame was reduced to just under two Gigabytes.

Created 9 Lag Features and implemented Rolling Features (5 rolling_mean features and 5 rolling_median features). This is implemented under the assumption that past data can influence or contain some intrinsic information about future data.

Data is prepared for the model by normalizing the data and label encoding using the Label Encoder where the labels are converted into numeric form.

C. Model Training

Now that our data is prepared and ready to be fed to the machine learning models, it is time to experiment with various models to determine which is the best one. In our case, we chose to experiment with Random Forest, AdaBoost, XGBoost, and LightGBM.

Random Forest: With the support of a lot of decision trees as well as a process called 'Bootstrap and Aggregation', also known as bagging, this algorithm is an ensemble practice capable of managing both classifications as well as regression techniques. This approach's core concept is combining several decision trees to get the outcome instead of being dependent solely on a single decision structure.

Various decision trees aid as the central learning models in this algorithm. We develop example datasets for each prototype by arbitrarily choosing rows and attributes from the data frame. This module is called Bootstrap.

Random Forest, to train our dataset, took an awful lot of time to train and in the end only provided a WRMSSE score of 0.667, which is worse than our target score of 0.65, even with the best parameters. This was not any different from AdaBoost, which took more time to train as compared to Random Forest. Especially given the fact that the data frame has over fifty-three million rows.

At this point, a machine learning regressor model with lesser training time was needed. So we explored XGBoost and LightGBM.

XGBoost: The gradient-boosted trees approach is widely used and well-implemented in open-source software called XGBoost. Gradient boosting is just a supervised learning process that combines the predictions of a number of weaker, simpler models to attempt to properly predict a target variable.

When experimenting with XGBoost, it was realized that the training time has drastically improved. More runs of the training have been performed because of the performance of XGBoost. With the right hyperparameters, the WRMSSE score turned out to be 0.636. This is a huge improvement from Random Forest.

While exploring more machine learning models to suit the project's use case, we have come across LightGBM, which appeared promising.

LightGBM: The two types of approaches used by LightGBM are 'Exclusive Feature Bundling' (EFB) and 'Gradient Dependent on Side Sampling' (GOSS). Therefore, GOSS will utilize only the collected information to determine the overall gain ratio and will actually omit the substantial amount of the data component that has minor gradients. The computation of information gain really gives more weight to data instances with large gradients. GOSS uses a smaller sample than similar models but still delivers reliable results with a large info gain.

Experimenting with the dataset with LightGBM yielded excellent results. With the right hyperparameters of 125 leaves and a learning rate of 0.075, a WRMSSE score of 0.615 was achieved. This is a huge improvement from the other models and much better than the project's target.

IV. Results and Experiments

As mentioned previously, different models were experimented with to achieve the best accuracy. And for each model, various parameters were experimented with to find the ideal parameters for the highest accuracy.

For Random Forest, we mainly changed 'n_estimators' and 'max_depth'. They are shown below with they WRMSSE scores.

Random Forest	n_estimators	max_depth	WRMSSE Score
Try 1	10	16	**0.679**
Try 2	10	100	**0.691**
Try 3	100	7	**0.676**
Try 4	100	21	**0.688**
Try 5	50	10	**0.667**

Figure 8: Random Forest experimental runs and results

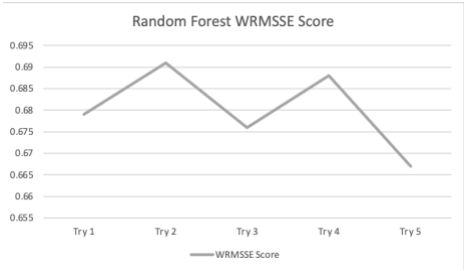

Figure 9: Random Forest experimental runs and results

For AdaBoost Regressor, we mainly changed 'n_estimators' and 'learning_rate'. The experimental findings are shown as below.

AdaBoost	learning_rate	n_estimators	WRMSSE Score
Try 1	1	50	**0.688**
Try 2	0.5	100	**0.664**
Try 3	0.3	75	**0.68**
Try 4	2	50	**0.676**
Try 5	0.3	50	**0.645**

Figure 10: AdaBoost experimental runs and results

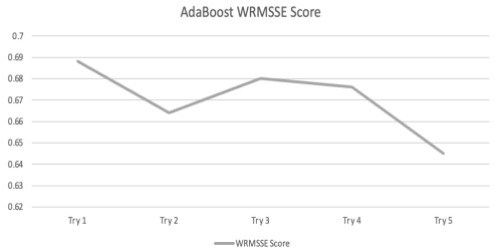

Figure 11: AdaBoost experimental runs and results

For XGBoost Regressor, we mainly changed ‘max_depth’ and ‘eta’. The WRMSSE scores for each trial are shown below:

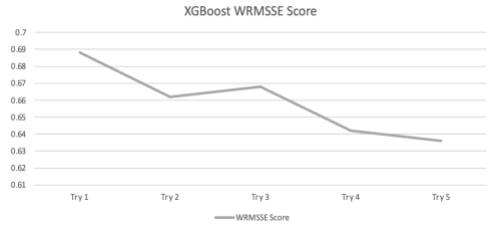

Figure 12: AdaBoost experimental runs and results

XGBoost	max_depth	eta	WRMSSE Score
Try 1	15	1	**0.688**
Try 2	25	0.75	**0.662**
Try 3	50	0.5	**0.668**
Try 4	5	0.3	**0.642**
Try 5	7	0.3	**0.636**

Figure 13: AdaBoost experimental runs and results

For LightGBM Regressor, we experimented with the parameters ‘leaves’ and ‘learning_rate’. The scores of LightGBM were the best compared to other models, all within the target score of 0.65 WRMSSE. The experiments are recorded below:

LightGBM	leaves	learning_rate	WRMSSE Score
Try 1	50	0.5	**0.623**
Try 2	75	0.75	**0.645**
Try 3	100	0.25	**0.648**
Try 4	20	0.1	**0.655**
Try 5	125	0.075	**0.615**

Figure 14: LightGBM experimental runs and results

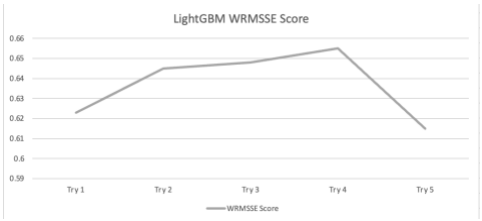

Figure 15: LightGBM experimental runs and results

Overall, LightGBM was the best model with respect to performance as well as accuracy. It takes the least amount of time to train and at the same time provides the best accuracy. This helped us run a greater number of experiments on it compared to other models.

V. Related Work

On the basis of the availability of pertinent historical data, studies have in the past been conducted to forecast sales for firms in the retail industry. (‘Walmart's Sales Data Analysis - A Big Data Analytics Perspective’, 2017) Several writers from the Fiji National University and The University of the South Pacific examined the Walmart dataset to forecast sales. The data was analyzed and visualized using tools including Distributed File Systems (Hadoop), Map Reduce platform, and Apache Spark as well as high-level programming environments like Scala, Java, and Python. Additionally, their analysis sought to determine whether the variables in the dataset had any bearing on Walmart sales.

In order to effectively manage resources, Harsoor and Patil (2015) forecasted Walmart Store sales utilizing big data applications such as Hadoop, MapReduce, and Hive. The sales data set utilized in this work was the same one used for analysis for this study, but Holt's winter algorithm was used to anticipate sales for the next 39 weeks. Tableau uses bubble charts to visually depict the anticipated sales.

Data scientist Michael Crown (2016) examined a comparable dataset but concentrated on using time series forecasting as well as non-seasonal ARIMA models to achieve his predictions. He used 2.75 years of sales data, including aspects of the store, date, weekly sales, department, and holiday data, to work on ARIMA modeling to produce one year's worth of weekly projections. The normalized root-mean-square error method was used to gauge performance (NRMSE).

Forecasting is no longer just for corporate growth. Many academics have attempted to use statistical analysis and machine learning to develop prediction models that can correctly forecast the weather, track stock prices and market movements, forecast patient ailments, etc.

Similarly to this, in 2017 Chouskey and Chauhan attempted to develop a weather forecasting program that accurately forecasts the weather and issues weather alerts to individuals and organizations so they may better prepare for unpredictable weather. The authors use MapReduce and Spark to build their models and collect data from various weather sensors. Since weather forecasts have an impact on every aspect of human life, the authors

used a variety of parameters, such as humidity, pressure, temperature, wind speed, etc., to produce more accurate predictions.

Rajat Panchotia (2020) used a different strategy to construct a predictive model utilizing linear regression, which sheds information on the different regression methodologies and the metrics that must be defined when building such models. He discusses the significance of outlining approaches that ought to be taken into account, such as examining the quantity of independent factors and the kind of dependent variables, figuring out the best fit, etc., based on the characteristics of the data and the most precise regression model that ought to be chosen results that were obtained.

It is essential to conduct a comparison examination of different models to make sure that the predictions are precise and that their application is not constrained. It is also vital for this study to test out numerous models because models behave differently depending on the type and volume of data.

VI. Conclusion

For the Walmart sales prediction, we considered the following Machine Learning techniques: Random Forest, AdaBoost, XGBoost, and LightGbm. We used dissimilar feature selections of many of these models as well as output WRMSSE including 28-day forecasts for each item in the raw data. When the four models are compared with each other, the LightGBM model showed the greatest accuracy, so our final score is of that model. Our score meets and surpasses our anticipation that the WRMSSE for LightGbm is around 0.615 when we had a target of going lower than 0.65. For better performance in the future, more data could be collected, both horizontally and vertically. A few more attributes describing each day with more detail would be much more helpful than just a few. With that, more precise models can be built, and finding the correlations would not be as difficult. To take this project one step further or to make it complete, an application can be developed that visually presents these predictions. These predictions can then be used by the Walmart staff to stock the products accordingly. This can help with the issues of overstocking or understocking the products. Overstocking can lead to more expired goods and understocking could make the customers less loyal to the super store.

VII. The Code

https://drive.google.com/file/d/1iycNkE2BXk147x0ggirJC2duF9u1VcET/view?usp=sharing

VIII. References

[1] Kaggle Competition: M5 Forecasting – Uncertainty, “https://www.kaggle.com/competitions/m5-forecasting-uncertainty”

[2] Chen, I. F., & Lu, C. J. (2017). Sales forecasting by combining clustering and machine-learning techniques for computer retailing. Neural Computing and Applications, 28(9), 2633-2647.

[3] Chen, T., Yin, H., Chen, H., Wu, L., Wang, H., Zhou, X., & Li, X.. (2018). Tada: trend alignment with dual-attention multi-task recurrent neural networks for sales prediction. IEEE International Conference on Data Mining (ICDM), Singapore, 2018, 49-58.

[4] Di Pillo, G. , Latorre, V. , Lucidi, S. , & Procacci, E. . (2016). An application of support vector machines to sales forecasting under promotions. 4OR, 14(3), 309-325.

[5] Harsoor, A. S., & Patil, A. (2015). Forecast of sales of walmart store using big data applications. International Journal of Research in Engineering and Technology eIS, 04, 51–59. https : / / doi . org / https : / / ijret . org / volumes / 2015v04 / i06 / IJRET20150406008.pdf

[6] Crown, M. (2016). Weekly sales forecasts using non-seasonal arima models. http : / / mxcrown.com/walmart-sales-forecasting/

[7] 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/january17/IJARCCE%2072.pdf