Predictive Sales Analysis and Forecasting for Walmart

Hamza Khan

August 8, 2024

Abstract

This report will discuss about the process of completing this research and predictive analysis project. Various types of machine learning techniques were applied to analyze the dataset, data processing, feature engineering, prediction models, and forecasting models. As well as a discussion of the journey of going through this whole process. In this report I will also go through the learning process me and my partner had in this fun experience of doing a data analyzing with machine learning as our jproject.

Contents

1 Introduction

- **Background and motivation**: The retail industry faces significant challenges in predicting sales due to the dynamic nature of consumer behavior. Accurate sales forecasting is crucial for inventory management, resource allocation, and strategic planning.

Our motivation was to get more experience related to our major and direction as a data and computer science. We wanted to do a project where we would learn, have fun because forecasting and using models felt like a fun project to do after researching about it. As well as wanting to incorporate machine learning topics in our project.

- **Objectives of the study**: The primary objective is to develop a robust predictive model to forecast sales for Walmart using historical sales data. This will help in understanding sales patterns and making informed business decisions.

  Our object was to developed a robust predictive model to forecast sales for retail stores and for it to be used on other stores. However our goal did change multiple times. It was finally changed to just forecasting Walmart sales due to running into issues such as having limited datasets, conducting wrong steps, long execution times, and confused on our approach. So our objective of the study was to mainly focus on Walmart sales trend.

- **Structure of the report**: The report is structured into several sections, such as data collection and preprocessing, exploratory data analysis, model selection and development, results and discussion, conclusion, and future work.

2 Data Collection and Preprocessing

- **Description of the dataset**: The dataset includes historical sales data from Walmart, comprising various features such as date, sales, store information, product details, quantity of orders, unit price, profit, and discount. It had many features which contributed to the idea of sales, the demand of products leading to the trend of Walmart in USA only. The dataset was about 300MB about more than 500,000 rows from 2019 to 2023. We chose to use a method from statistics called stratified sampling and only got 1000 samples for prediction and analyzing.

- **Dataset from data.world**:

  Link: `https://data.world/ahmedmnif150/walmart-retail-dataset`

- **Data cleaning and preprocessing steps**:

  Data cleaning involved handling missing values, correcting data types, dealing with outliers.

- **Handling missing values and outliers**: Missing values were imputed using appropriate statistical methods, and outliers were treated based on domain knowledge, Interquartile Range method (IQR) to remove outliers. Removed data for 2023 after seeing it was less compared to 2019-2022. After doing calculations we validated the features to check for negative values and use absolute mathematics method to make it positive for our

predictions. Made sure to make our data look as much as clean as it can and made sense

- **Feature engineering and selection**: New features were created to capture seasonality and trends. Features were selected based on their correlation with the target variable. Main focus was sales, to look for the demand of products, using every features that related to the demand. feature engineered some features such as created log sales and log prices. engineered estimated sales using every feature contributing to the demand with linear combination or linear regression equation.

3 Exploratory Data Analysis (EDA)

- **Initial data exploration**: The initial exploration included summary statistics and visualizations to understand the distribution and relationships between variables or features. Created histograms, bar graphs, correlation heatmaps for checking the customers age, top 5 most wanted products, yearly sale, relationship between each features and more.

- **Visualization of key variables**:

  Key variables such as sales over time and sales by product, different types of feature engineered sales, category were visualized using line plots, bar charts, and histograms. Attempted to use the kernel trick, to see more visualization.

- **Summary statistics**:

  Statistics provided insights into central tendencies, dispersion, and overall data quality. As well as the visualization graphs.

4 Model Selection and Development

- **Selection of machine learning models**:

  Various models including Linear Regression, Decision Trees, Random Forests, Elastic, Lasso, Ridge, Neural Network and Gradient Boosting were used for the analysis. We did about 20-40 testing, experimenting, execution time, and ran into many issues. We wanted to use the ensemble method, to mix all of these models, and make a efficient powerful model.

  We also used other forecasting models, Arima, LSTM, XGBoost, and Prophet. These we time series forecasting models we implemented and test. We did try to attempt to mix these models to experiment but we not able to due to complications.

- **Model training and validation**:

  Models were trained using a stratified shuffle split to ensure balanced representation of the data in training and validation sets as mentioned.

We used standardizing/normalizing during each model we were using and not during data preprocessing stage or else it would have been done twice. We did make this mistake and later fixed it.

- **Hyperparameter tuning**:

  Grid search, Randomized Search and cross-validation techniques were used to optimize hyper-parameters for each model. As well as using pipeline and param-grid were used which has specific needs for each model. Pipeline was using to streamline process of performing sequence of data and param-grid was used to define the hyper-parameters and tune for each model.

- **Model evaluation metrics**:

  Models were evaluated using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). The (R2) evaluation gave the percentage for training and testing telling us about under-fitting, generalization, and over-fitting. After seeing if the model and the final ensemble model is over-fitting or under-fitting we continuously made many changes to try to have as much as a less over-fitting/under-fitting and more generalization.

5 Results and Discussion

- **Performance of different models**:

  The performance of various models was compared based on evaluation metrics. Random Forest and Gradient Boosting showed promising results. But so did linear regression and few others which was a surprised since we were expecting only Random Forest and Gradient Boosting to be the best since we were using a complex approach.

  Together using ensemble method we had the best results, .99 for both training and testing for (R2). And an MSE of 20.5 for training and for testing 343.5, which is the MSE is not as good as we expected. The best result we had was 20.9 MSE for training and 54.3 for testing, and same (R2) as before however we lost this better result and weren't able to replicated it so we used the second best result.

  We used this to do forecasting, predicting the future values mainly for 2023 to 2026 however we also did a graph of looking at predicted estimated sale and true sales to which to our surprised the predicted sales on the historical true sales was fitting very perfectly. This did put us in doubt but we checked our code over and over again and didn't see anything out of the ordinary. We were able to see trends for the future.

  The 4 forecasting models we used, only Prophet seemed to make the most sense in the terms of correctness in future forecasting trends. Arima, LSTM, and XGB did give us graphs however interpreting wasn't easy and didn't really match each other. The specific numbers didn't really match

one another however we were able to see how much drops in trends and how much ups there were and able to see it kind of matches each other.

Prophet and ensemble method kind of was similar only for the first year of forecasting that matched and Prophet had the best visualization techniques. We were also able to see which days of the weeks sales are more higher.

Overall for the forecasting we did make interpretations but we are in a bit of a doubt about its correctness. We were not able to really find Walmart forecasting trends on the web to see if it matches with ours. We also came to a realization that the most wanted product from Walmart are home appliances, excluding other products from Walmart.

- **Comparative analysis**:

  A comparative analysis highlighted the strengths and weaknesses of each model. Ensemble methods generally outperformed individual models but sometimes. Because we noticed Linear Regression especially was out performing sometimes compared to other models. We hypothesized which models would be the best such as Random Forest but in some case we were wrong.

- **Insights from the results**: The analysis revealed significant seasonal patterns and the impact of promotional activities on sales as well.

- **Limitations of the study**: The study's limitations include the availability of data, assumptions in model training, and the potential for overfitting.

  There were alot of mistakes we made and issues we struggled through. We spent most of the time testing, finding correct dataset, changing our approaches/goals, waiting for the execution time to end so we can change our code if the results weren't the best, realizing our mistakes too late, and coding, and researching.

  We were set with our initial goal to make a predictive sale analysis that can be used for datasets, however we realized it might not be reasonable or possible due to complex or variety in datasets. We also only found one dataset that seemed the best to use. We were also spending time to find only datasets that has sales but we realized later on that we can use feature engineering to use other features to make estimated sales.

  We only used 1000 random stratified samples and not more than this due to the execution timings and other complications.

  We used Google Colab for our coding, and used allot of researching resourcing to understand how to do each step. Google Colab did limit us in someways such as in computation units, and GPU. We had issues in executing some areas of code because it took about 4 hours or more. It took us a while to realize we can use different GPU or create copies to restart execution.

6 Conclusion and Future Work

- **Summary of findings**:

  The study successfully developed predictive models that can forecast sales with a reasonable accuracy. This study was also able to give an analysis as well.

- **Implications for Walmart**: The insights gained can help Walmart optimize inventory, plan promotions, and make strategic decisions.

- **Learning**: We learned alot from this project. From data collection, cleaning, processing, to using these important models/tools, and learning from our mistakes. It was really fun project to do over this summer, and we hope to make it more better. We learned through the slides on canvas on these concepts, and through researching on the web. Predicting and forecasting future values always sounded very fun especially when using these probability and statistics concept in coding, seeing how mathematics has a strong foundation and relation in this area. We also coded step by step, block by block to go through the difficult proces slowly.

- **Suggestions for future research**: Future research could explore more advanced models like deep learning, incorporate external data sources, and improve feature engineering techniques. We also plan to learn from our mistakes and make more enhancements to our project.

7 References

1 GeeksforGeeks. Exploratory Data Analysis in Python. Retrieved from `https://www.geeksforgeeks.org/exploratory-data-analysis-in-python/`

2 GitHub. EDA Notebooks. Retrieved from `https://github.com/topics/exploratory-data-analysis`

3 Stack Overflow. EDA Techniques. Retrieved from `https://stackoverflow.com/questions/tagged/exploratory-data-analysis`

4 GeeksforGeeks. Stratified Sampling in Pandas. Retrieved from `https://www.geeksforgeeks.org/stratified-sampling-in-pandas/`

5 GitHub. Stratified Sampling Code. Retrieved from `https://github.com/search?q=stratified+sampling`

6 Stack Overflow. Stratified Sampling. Retrieved from `https://stackoverflow.com/questions/tagged/stratified-sampling`

7 GeeksforGeeks. Regression Analysis in Machine Learning. Retrieved from `https://www.geeksforgeeks.org/regression-analysis-in-machine-learning/`

8 GitHub. Regression Analysis. Retrieved from https://github.com/topics/regression-analysis

9 Stack Overflow. Regression Models. Retrieved from https://stackoverflow.com/questions/tagged/regression

10 GeeksforGeeks. Time Series Analysis. Retrieved from https://www.geeksforgeeks.org/time-series-analysis/

11 GitHub. Time Series Analysis. Retrieved from https://github.com/topics/time-series-analysis

12 Stack Overflow. Time Series Analysis. Retrieved from https://stackoverflow.com/questions/tagged/time-series

13 GeeksforGeeks. ARIMA Model Time Series Forecasting. Retrieved from https://www.geeksforgeeks.org/arima-model-time-series-forecasting/

14 GitHub. ARIMA Model. Retrieved from https://github.com/topics/arima

15 Stack Overflow. ARIMA Model. Retrieved from https://stackoverflow.com/questions/tagged/arima

16 GeeksforGeeks. Long Short-Term Memory Networks Explanation. Retrieved from https://www.geeksforgeeks.org/long-short-term-memory-networks-explanation/

17 GitHub. LSTM Implementation. Retrieved from https://github.com/topics/lstm

18 Stack Overflow. LSTM. Retrieved from https://stackoverflow.com/questions/tagged/lstm

19 GeeksforGeeks. XGBoost. Retrieved from https://www.geeksforgeeks.org/xgboost/

20 GitHub. XGBoost Code. Retrieved from https://github.com/topics/xgboost

21 Stack Overflow. XGBoost. Retrieved from https://stackoverflow.com/questions/tagged/xgboost

22 GeeksforGeeks. Forecasting with Prophet. Retrieved from https://www.geeksforgeeks.org/forecasting-with-prophet/

23 GitHub. Prophet Implementation. Retrieved from https://github.com/topics/prophet

24 Stack Overflow. Prophet. Retrieved from https://stackoverflow.com/questions/tagged/prophet

25 Stack Overflow. Example of sales prediction using machine learning in Python. Retrieved from https://facebook.github.io/prophet/

A Appendix

- **Additional figures and tables**: Additional visualizations and tables that provide deeper insights into the data.

Model	**Training MSE**	**Testing MSE**	**Training R^2**	**Testing R^2**
Lasso Regression	0.4399621278	3.3189848872	0.9999992245	0.9999933602
Random Forest	143.9248826342	4838.8292807894	0.9997463171	0.9903197451
Gradient Boosting Model	5574.9082003059	4134.8262450875	0.9901736321	0.9917281289
Decision Tree	0.8602539747	3586.6020339802	0.9999984837	0.9928248715
Neural Network	951.6560332385	11892.8469081357	0.9981953369	0.9805725516
Linear Regression	1.2154432126	1.110342550026471	1.0000000000	1.0000000000
Elastic Net	4.6861899849	2.6315806283	0.9999911134	0.9999957012
Ridge Regression	1.1381866	1.388515542	1.0000000000	1.0000000000
Ensemble Model	20.9840085503	54.3119011067	0.9999602072	0.9999112793

Table 1: Model Performance Comparison

Model	**Training MSE**	**Testing MSE**	**Training R^2**	**Testing R^2**
Lasso Regression	0.4399621278	3.3189848872	0.9999992245	0.9999933602
Random Forest	143.9248826342	4838.8292807894	0.9997463171	0.9903197451
Gradient Boosting Model	5574.9082003059	4134.8262450875	0.9901736321	0.9917281289
Decision Tree	0.8602539747	3586.6020339802	0.9999984837	0.9928248715
Neural Network	2345.0347563957	10289.5367098171	0.9958666272	0.9794154056
Linear Regression	5.951888742	7.3544236781	1.0000000000	1.0000000000
Elastic Net	155.8785344817	203.1706101532	0.9997252475	0.9995935498
Ridge Regression	1.17254906	2.389266	1.0000000000	1.0000000000
Ensemble Model	20.5206590949	343.5874968719	0.9999638302	0.9993126406

Table 2: Model Performance Comparison 2

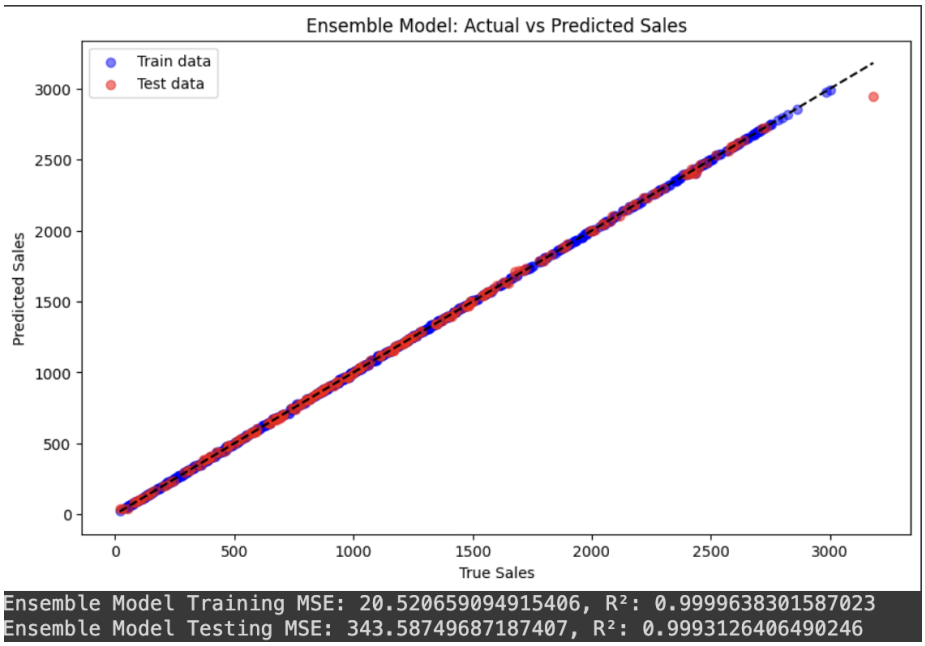

Figure 1: Ensemble Mode

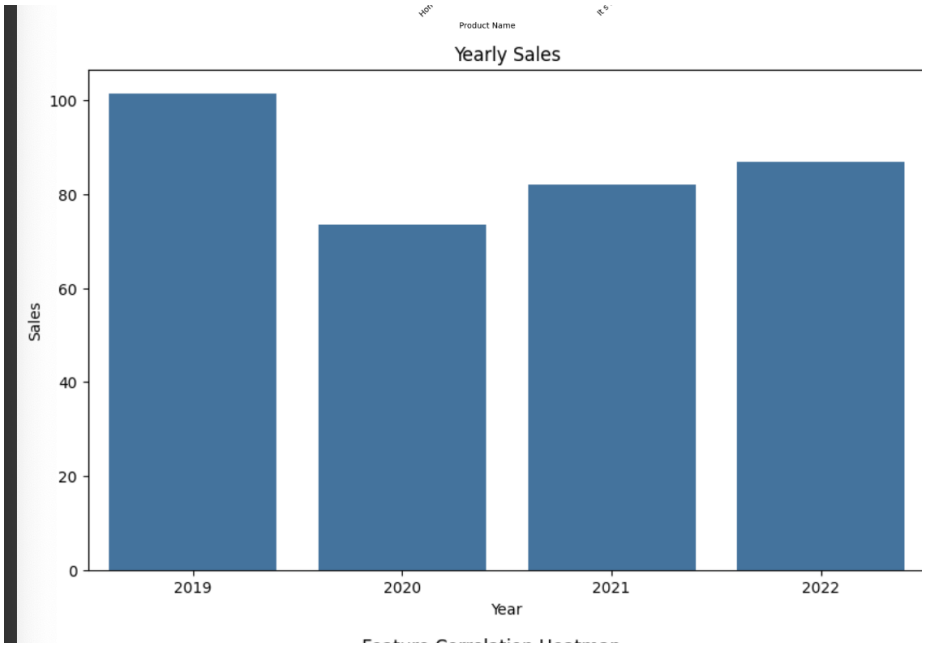

Figure 2: Yearly Sales

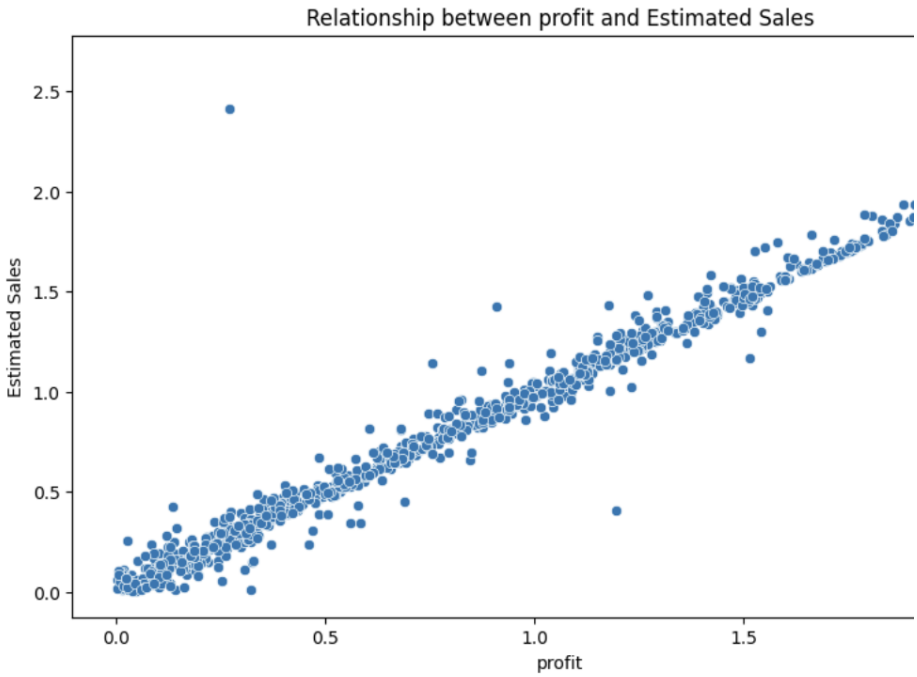

Figure 3: Relation between profit and Estimated Sales

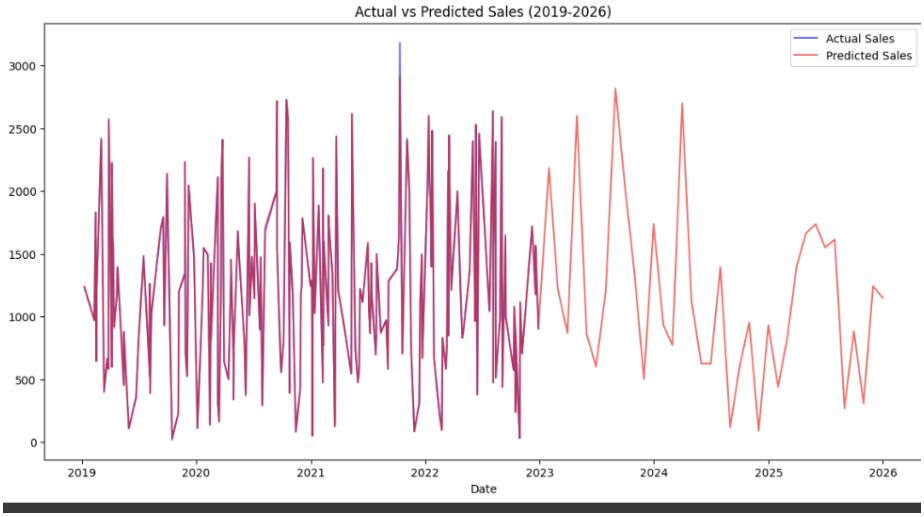

Figure 4: Forecast Prediction on Historical Sales and Future

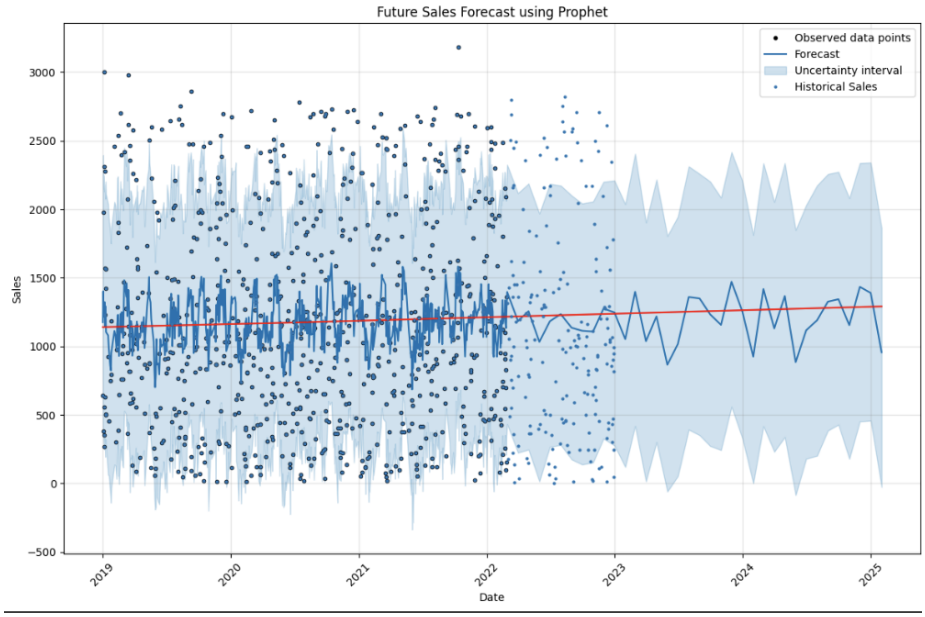

Figure 5: Prophet Forecasting