End-to-End Retail Insights: XGBoost-Driven Sales Forecasting, Promotion Effectiveness, and Volatility Analysis

Team Members: Dean Carpenter, Rohan David, Sebastian Martinez, Simoni Dalal, Sreejony Sengupta

Abstract

In the low-margin world of retail, the difference between profit and loss often comes down to inventory management. Overstocking leads to wasted capital and markdowns, while understocking leads to missed revenue and dissatisfied customers. This project tackles the challenge of predicting weekly sales across numerous Walmart stores and departments. By leveraging advanced feature engineering and Gradient Boosting (XGBoost), we achieved a model with an R2 score of 0.95, specifically optimized to minimize the Weighted Mean Absolute Error (WMAE)—a metric that prioritizes accuracy during high-stakes holiday weeks. Beyond prediction, we extended our analysis to business applications, including promotion effectiveness and Revenue at Risk (VaR) assessment.

The Problem: The High Cost of Volatility

Retail sales are not linear; they are highly seasonal and driven by erratic events like Super Bowl Sunday, Thanksgiving, and Christmas. Standard forecasting often fails to capture these spikes. The business problem is financial: accurate forecasts are required to optimize labor planning and cash flow. Crucially, errors made during holiday weeks are exponentially more damaging than errors in a standard week.

The Approach

While traditional time-series models (like ARIMA) focus on a single trend, our approach frames forecasting as a supervised regression problem using Machine Learning. Our key contributions include:

1. **Feature Engineering:** Constructing lag and rolling window features to capture temporal dependencies without using recurrent neural networks.
2. **WMAE Optimization:** Evaluating models based on a weighted metric that penalizes holiday errors 5x more than standard errors.
3. **Financial Risk Analysis:** Moving beyond accuracy to assess Cash Flow Volatility and Revenue at Risk (VaR) for different store types.

Data Collection & Description

Source: The dataset is sourced from the **Walmart Recruiting - Store Sales Forecasting** Kaggle competition.

Characteristics:

- **Scale:** Historical sales data for 20 stores with 16 departments each.
- **Granularity:** 3 years of data at a weekly level for each store-department combination
- **Target Variable:** Weekly Sales (i.e. to predict weekly sales in $ at the store-department level)
- **Store Attributes:** Store Size and Store Type (A, B)
- **Holiday/Regular week**: A TRUE/FALSE indicator for a holiday/regular week
- **Promotional:** Anonymized Markdown data (Markdown 1-5) representing various promotional events
- **Macroeconomic Features:** Consumer Price Index (CPI), Unemployment Rate, Fuel Price and Temperature

Data Pre-Processing and Exploration

Raw data alone is rarely predictive enough for complex time-series tasks. We engineered features to help non-temporal models "see" time:

- **Lag Features:** We created *lag_1_sales* and *lag_52_sales* (Last Year, Same Week) to capture immediate trends and yearly seasonality.
- **Rolling Statistics:** We calculated *rolling_mean_4w* and *rolling_std_4w* to smooth out noise and capture recent volatility.
- **Date Decomposition:** We extracted *WeekOfYear, Month,* and *Quarter* to allow the model to learn seasonal cycles.

Code Snippet: Creating Rolling Features

Python

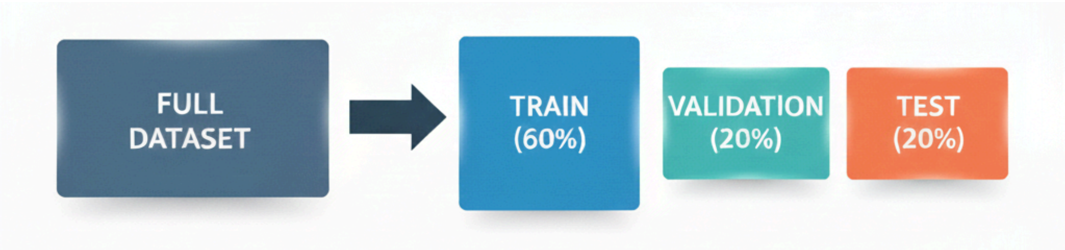
```
# Example of rolling window feature engineering
df['rolling_mean_4w'] = df.groupby(['Store', 'Dept'])['Weekly_Sales'] \
   .transform(lambda x: x.shift(1).rolling(window=4).mean())
```

Next, through exploratory data analysis, we found that Markdowns (promotional features) contained significant missing values, which we imputed with 0 (assuming null implies no promotion). Correlation analysis revealed that Store Size and Store Type were strongly correlated with Sales, whereas macroeconomic factors like Unemployment showed surprisingly weak linear correlations, suggesting their impact is likely non-linear or threshold-based.

Learning/Modeling

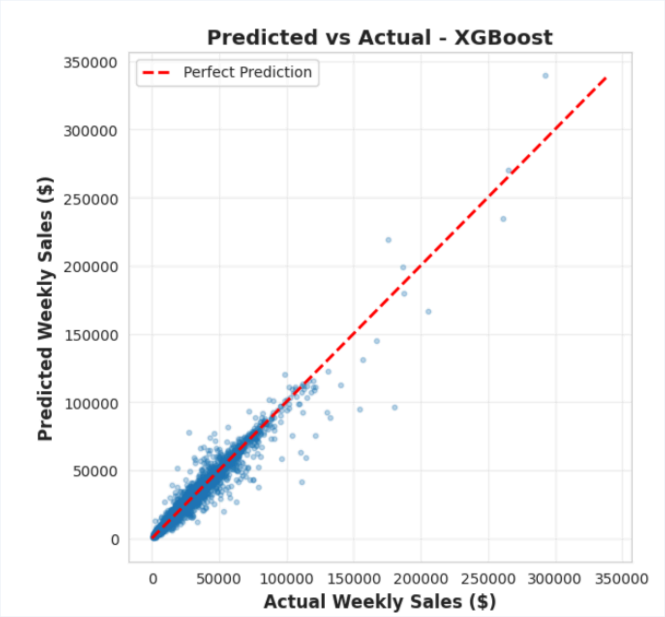
Now, to answer the most critical question: which algorithm would best forecast weekly sales? We trained and compared five different machine learning models. The aim is to find the best balance between accuracy and generalization.

To ensure a fair analysis, we split our data into a train-validation-test set. And for evaluating model performance, we are using these four metrics: MAE (average dollar error), RMSE (penalizes large errors), R² (variance explained), and WMAE (a metric that weights holiday weeks 5x more).

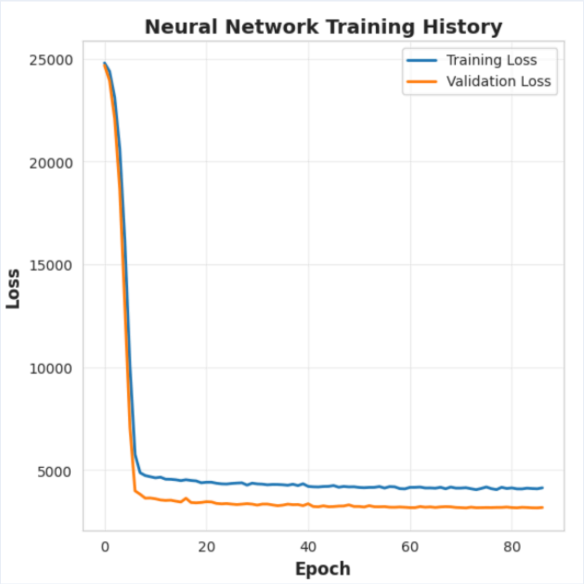


We started with a simple linear model- the ridge regression, that served as our baseline to establish the minimum acceptable performance. Even a basic model captured 82% of sales patterns, validating the feature engineering we did. However, because linear models can't capture complex interactions between promotions, seasonality and economics, we tried the Random Forest Regressor. With 300 decision trees and bootstrap sampling, we definitely saw a significant improvement over the Ridge model. The Random Forest model handled non-linearities well but struggled to extrapolate trends outside the training range.

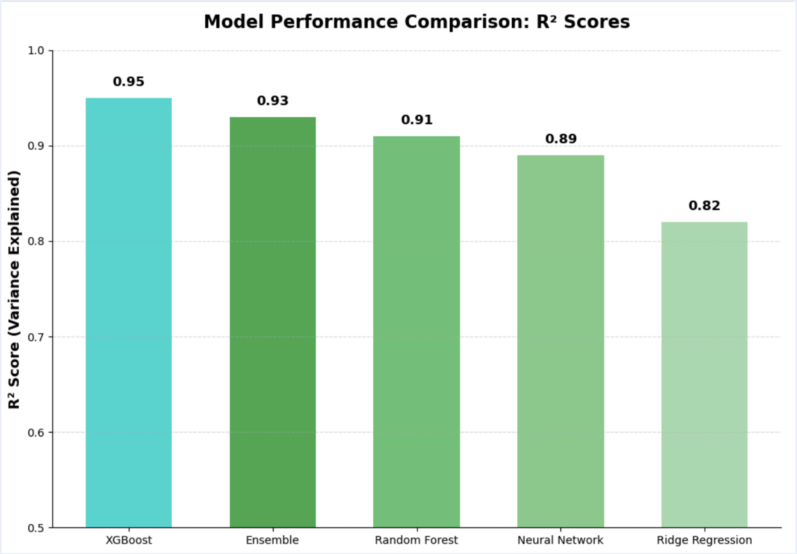
Next, we explored XGBoost with 500 trees and a learning rate of 0.05. We added regularization to prevent overfitting. The model achieved a 95% R² and $2,822 WMAE which represents approximately an error of 15% which is good for retail forecasting given the inherent volatility of the industry.



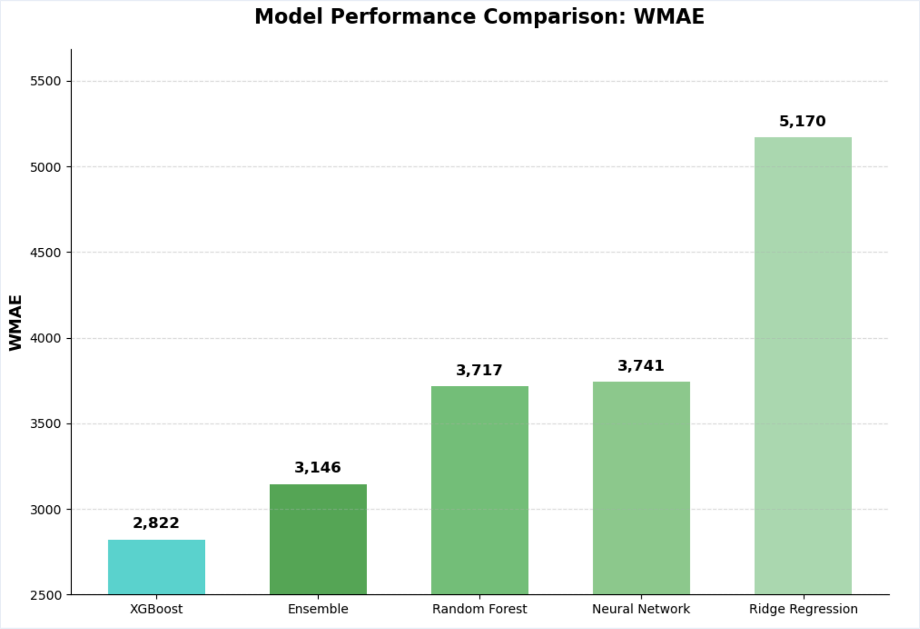
Next, we explored deep learning, by building a Neural Network with 4 hidden layers, batch normalization and dropout for stability. We configured Adam optimizer with learning rate scheduling and huber loss which is robust to outliers. Looking at the output, the neural network underperformed compared to tree-based models. This was expected, since deep learning excels at data that contains images, text and audio but struggles with small to medium tabular datasets (in our case just having 42k+ rows of data).



Next, we wanted to know the combined predictions from Random Forest, XGBoost, and Neural Network by building an Ensemble Model, with weights based on validation performance (39% XGBoost, 31% RF, 30% MLP). Surprisingly, the ensemble **didn't beat XGBoost alone**. This is because when one model dominates (like XGBoost in this case), averaging it with weaker models dilutes performance rather than improving it. Ensembles work best when multiple models are similarly strong—that wasn't our case.



XGBoost emerged as the clear winner! Why did it work? The boosting mechanism corrected the errors of previous trees, allowing it to "learn" the specific spikes of holiday weeks better. We have used this model for all subsequent business analyses.



Promotional Analysis

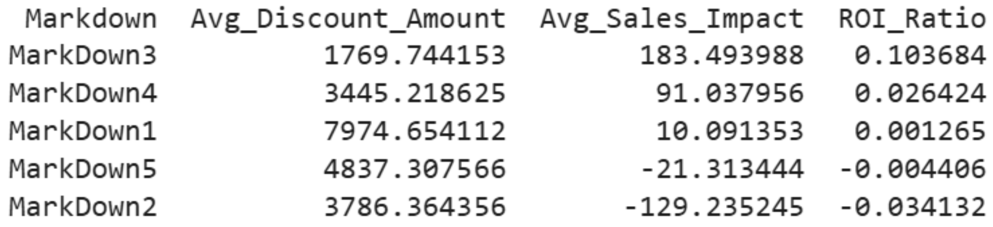
We analyzed the impact of the promotional features (Markdowns 1-5) on weekly sales (Weekly_Sales). This was done using SHAP values of the promotional features obtained from the XGBoost model.

The SHAP values of the features were preferred over the Feature Importance approach as the SHAP approach assigns a contribution value to each feature for every prediction. This helps to estimate the sales impact of each of the promotional features.

Feature	SHAP Value
Markdown3	146.5025
Markdown4	108.3508
Markdown2	91.0053
Markdown1	41.6314
Markdown5	40.6741

These were the global SHAP values of the Promotional Features from the dataset. The global values help in understanding the predictive power of each of the promotional features.
It can be noted that Markdown3 had the highest predictive power on Weekly Sales whereas Markdown5 had the lowest predictive power on Weekly Sales.

Using SHAP values, the Average Sales Impact was calculated to arrive at the ROI for the Promotional Features.



```
  Markdown  Avg_Discount_Amount  Avg_Sales_Impact  ROI_Ratio
 MarkDown3          1769.744153        183.493988   0.103684
 MarkDown4          3445.218625         91.037956   0.026424
 MarkDown1          7974.654112         10.091353   0.001265
 MarkDown5          4837.307566        -21.313444  -0.004406
 MarkDown2          3786.364356       -129.235245  -0.034132
```

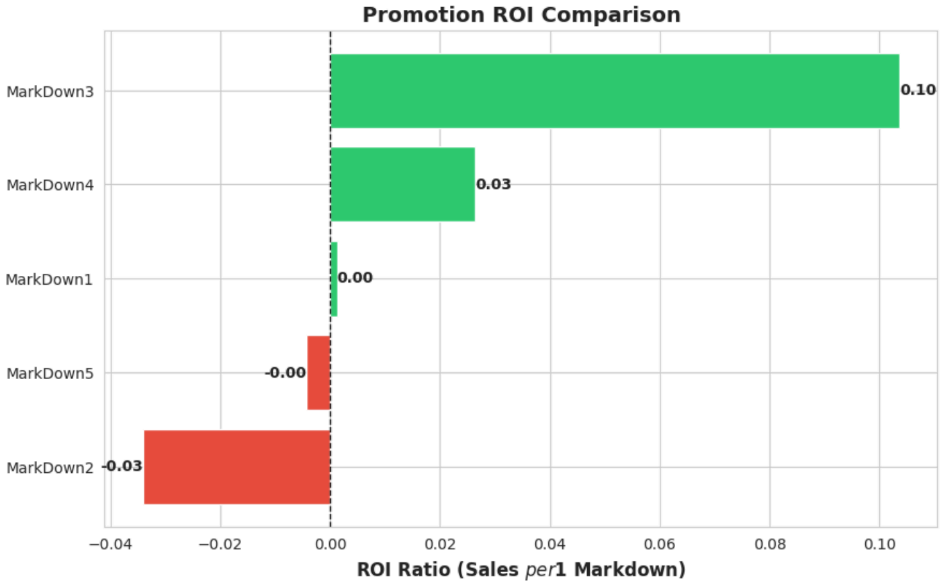
The **Avg_Discount_Amount** was the average amount spent on the markdown feature (the average value of the promotional feature across the dataset)

The **Avg_Sales_Impact** was the mean SHAP value of the promotional feature which helps to understand the positive or negative impact on weekly sales.
(These values are different from the global SHAP values for the feature as the global SHAP values take the absolute average of SHAP values across the whole dataset. The

Avg_Sales_Impact take the average of SHAP values across the dataset only where there are entries for the Markdown features)

The **ROI_Ratio** was obtained by dividing the **Avg_Sales_Impact** by **Avg_Discount_Amount** for the promotional features.

The ROI can be interpreted as the Sales generated per $ spent on markdown.
For instance, the ROI of Markdown3 was 0.1036. So, for every $1 spent on Markdown3, there was an increase in weekly sales by 0.1036.



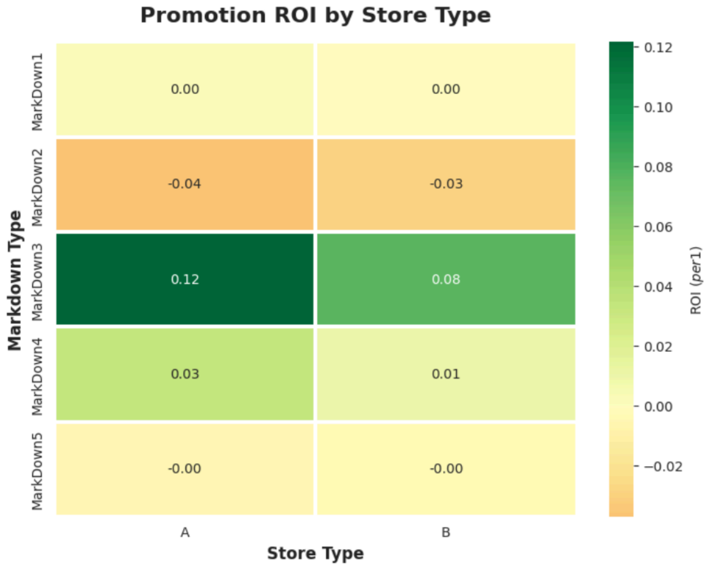
From this visual, it can be noted that **Markdown3, Markdown4 and Markdown5** had a **positive ROI** whereas **Markdown5 and Markdown2** had a **negative ROI.**

From the ROI analysis of promotional features across the dataset

- Increase investment in Markdown3 and Markdown4 as these had the highest ROI
- Eliminate or reduce investment in Markdown5 and Markdown2 as these had the lowest ROI.

ROI Analysis of Promotions by Store Type

We also analyzed the **ROI of the promotional features** at the **Store Type** Level as the dataset had **Store Types A and B.**



The above visual shows the ROI values of the 5 promotional features for store types A and B.

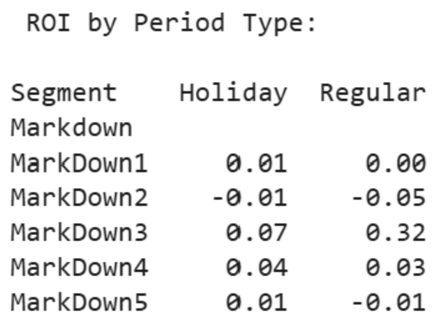
- **MarkDown3** had the **highest ROI** across both **store types A and B** with **ROI values of 0.12 and 0.08 respectively.**
- **MarkDown2** had the **worst ROI** across both the **store types A and B** with **ROI values of -0.04 and -0.03 respectively.**

From this analysis, **Markdown2** can be **phased out** to save on promotional expenses and the budget for **Markdown3** can be **increased** as it had the **highest ROI** across both store types. This insight helps the retailer to plan and allocate appropriate budgets for different promotions across different store types to generate savings and maximize revenue.

ROI Analysis of Promotions during Regular and Holiday Weeks

Next, we analyzed the **ROI of promotions** during **Regular and Holiday weeks**. The dataset had 10 Holiday weeks and 133 Regular Weeks.

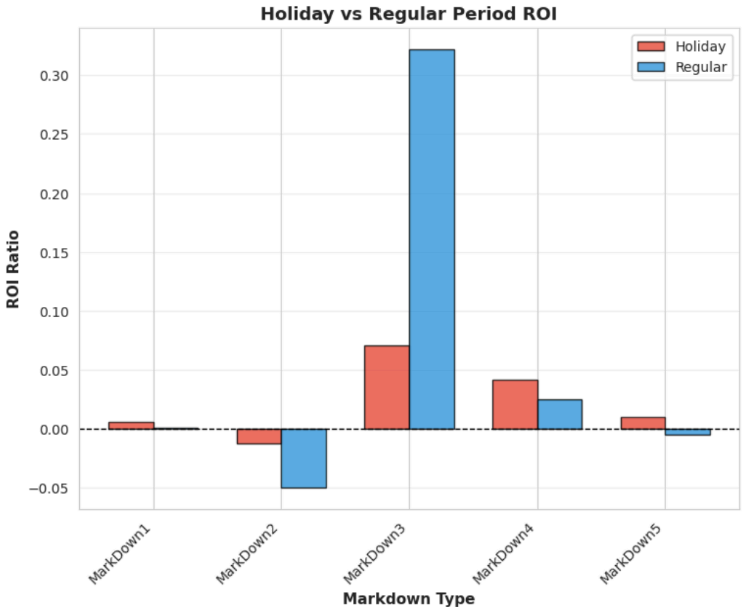
The Holiday weeks were the weeks surrounding Super Bowl, Labor Day, Thanksgiving and Christmas.



```
 ROI by Period Type:

Segment    Holiday  Regular
Markdown
MarkDown1     0.01     0.00
MarkDown2    -0.01    -0.05
MarkDown3     0.07     0.32
MarkDown4     0.04     0.03
MarkDown5     0.01    -0.01
```

The above snippet shows the ROI of the 5 promotional features across the Holiday and Regular weeks



From the visual, it can be observed that though **MarkDown3 has a positive ROI during Holiday and Regular weeks, the ROI during Holiday weeks (0.07) showed a decline of about 78% from the ROI during Regular weeks (0.32).**

- **MarkDown4** had a **higher ROI** during the **Holiday weeks (0.04)** compared to **Regular weeks (0.03)**. Surprisingly, **Markdown5 had a positive ROI during the Holiday weeks.**
-
- **During Holidays, the retailer can increase the budget allocation for Markdowns 4 and 5 while reducing allocation for Markdown3.**

Though the ROI analysis of promotional features at a global level is required, it is pertinent to identify which promotional features have higher ROI depending on store type and holiday/regular weeks to maximize the sales revenue and savings for the retailer.

Sales Volatility & Revenue-at-Risk Analysis

In this analysis, we examined week-to-week sales behavior across stores and departments using predictions from the XGBoost model. By calculating volatility metrics—including the coefficient of variation (CV), sales ranges, and stability classifications—we were able to identify how predictable or unpredictable cash flow is at the store level.

The results show that 60% of stores fall into the “Volatile” category, meaning their weekly sales fluctuate widely. Even the most stable stores still sit in the “Moderate” volatility range, indicating that while some locations are more manageable than others, the mass retail business operates in an environment with meaningful demand variation. These patterns provide early warning signals for store managers and finance teams where operational risk is inherently higher.

We also evaluated Revenue at Risk using Value-at-Risk (VaR) calculations to quantify the worst-case sales outcome. Stores with the highest downside risk, such as Stores 14, 18, and 4, could experience sales drops of more than 80% below their average during their weakest weeks. This makes them especially vulnerable in cash-sensitive periods and highlights where contingency planning, staffing flexibility, and inventory buffers become essential. Interestingly, store types also reveal a trade-off: Type A stores generate significantly higher revenue but show greater volatility and downside exposure, while Type B stores are smaller but slightly more predictable.

Together, the volatility and VaR insights allow leaders to pinpoint which stores are stable profit engines versus which require closer monitoring and more dynamic management for smooth operations.

Error Analysis: Where did we fail?

We drilled down into the errors to understand the business implications.

1. **Categorical Error:** We found that **Department 5** and **Stores 13, 20, and 10** contributed disproportionately to the error. These are likely seasonal or "clearance" departments with erratic behavior or stores with unpredictable traffic.
2. **Holiday Weighting:** Despite our focus on WMAE, the model still showed a higher average error during holiday weeks compared to non-holiday weeks. This indicates that while we captured the trend, the *magnitude* of the Christmas spike is still difficult to predict perfectly.

Business Insights

- **Promotional Analysis :** From promotional analysis at a global level, by store type and by holiday/regular periods, we found that certain promotions can be prioritized over the other. The prioritization approach should be dynamic in nature to maximize sales revenue and savings.

- **Macroeconomics:** Through elasticity analysis, we found that Fuel Price and Unemployment have a negligible impact on short-term weekly sales. This suggests customers treat Walmart as an essential retailer, resilient to minor economic shifts.
- **Cash Flow Risks:** Our VaR (Revenue at Risk) analysis identified **Type A** stores as high-volume but high-volatility. We recommend holding higher safety stock for these specific locations compared to **Type B** stores.

Conclusion

Summary

This project demonstrated that with robust feature engineering (specifically lags and rolling windows), a gradient boosting model (XGBoost) can predict retail sales with 95% R^2. We successfully transitioned from raw data to a financial risk assessment tool.

Lessons Learned

- **Feature Engineering > Model Complexity:** A well-engineered XGBoost model outperformed a deep neural network (MLP). The "smarter" model was the one with better data, not more layers.
- **The "Black Box" Problem:** While XGBoost is accurate, explaining *why* sales dropped to a store manager requires tools like SHAP values, which we identified as a necessary next step.

Future Work

1. **Prophet & LSTMs:** To better capture the holiday effects, we plan to implement Facebook Prophet (for its additive seasonality) and LSTMs (for sequence modeling).
2. **Safety Stock Optimization:** We intend to convert our RMSE scores into actual inventory dollar recommendations to quantify savings.

References & Links

- **Dataset:** Kaggle Walmart Recruiting - Store Sales Forecasting
- **Key Libraries:** Scikit-Learn, XGBoost, Pandas, Seaborn.
- Analytics Vidhya. (2021). "Complete Guide to Parameter Tuning in XGBoost with Codes in Python."
- DataCamp. (2023). "Introduction to SHAP Values for Machine Learning Interpretability." DataCamp Tutorial.
- Koehrsen, W. (2019). "Explaining Your Models with the SHAP Values." Towards Data Science.