CS 559: Data-Driven Sales Forecasting and Pricing Strategy Optimization with Machine Learning

Arman Singh
Stevens Institute of
Technology
asingh102@stevens.edu Fall
2024

Abstract

This project focuses on leveraging machine learning models to forecast sales using 2023 data from the Black Friday Sales dataset. The primary goal is to develop predictive models that help optimize pricing strategies, improve inventory management, and support revenue growth. Various machine learning algorithms are implemented and compared to determine their efficacy in predicting sales with accuracy. This report documents the methodology, results, and findings of this experiment.

1 Introduction

Sales forecasting plays a crucial role in retail business operations, enabling businesses to efficiently allocate resources, manage inventory, and design effective pricing strategies. Traditional statistical models for forecasting often fail to capture the complexity of modern retail data sets influenced by factors such as demographics, promotions, and market trends. This project utilizes machine learning techniques to address these challenges, leveraging Kaggle's Black Friday sales dataset to train and test multiple predictive models.

Objectives:

- Implement machine learning models to forecast sales.
- Compare models based on performance metrics to identify the most suitable approach.
- Extract actionable insights to optimize retail operations.

2 Related Work

Previous studies have explored statistical models such as linear regression for sales forecasting. Although straightforward, these models often lack the adaptability required for modern retail environments. In contrast, machine learning methods such as Random Forests, Gradient Boosting, and Neural Networks offer the ability to capture complex data relationships. However, these advanced models have limitations, including risks of overfitting and high computational demands. **Notable Contributions**:

- **Gradient Boosting Algorithms**: Research by Hitesh S.M and Yukthi A emphasizes the effectiveness of the Gradient Boosting Algorithm in uncovering insights within large datasets, thereby improving forecast accuracy and efficiency. Their study identifies this algorithm as the optimal model for predicting future sales trends, demonstrating superior accuracy compared to traditional methods. https://www.ijnrd.org/papers/IJNRD2401051.pdf

- **XGBoost Regressor**: XGBoost Regressor: A study focusing on the retail sector, particularly Big Mart, utilized the XGBoost Regressor algorithm for sales prediction. The research underscores the importance of data preprocessing, analysis, and model training in accurately forecasting sales, which aids in inventory management and strategic decision-making. https://www.jetir.org/papers/JETIRGH06010.pdf
- **Hybrid Models**: Hybrid Models: Another approach integrates multiple machine learning models to enhance predictive performance. For instance, combining LightGBM and XGBoost frameworks has been shown to improve sales forecast accuracy by leveraging the strengths of both models. https://ieeexplore.ieee.org/document/9424806

This project aims to evaluate these methods in the context of retail sales forecasting, comparing their performance to traditional linear regression and identifying the best approach for complex retail environments.

3 Methodology

3.1 Problem Formulation

The problem is formulated as a supervised learning task, where the goal is to predict purchase amounts based on historical sales data. The dataset is treated as a regression problem with various customer and product-related features serving as inputs, and the target variable being the purchase amount.

3.2 Dataset

- **Source**: Black Friday Sales dataset from Kaggle.
- **Size**: Over 500,000 records, representing a wide variety of customer transactions.
- **Features**:
  - **Customer Demographics**: Attributes such as age, gender, marital status, and years spent in the current city.
  - **Product Details**: Features like product ID, category, and subcategory information.
  - **Transaction Information**: Purchase amount, frequency of transactions, and the city category of the purchase location.

3.3 Data Processing

Data preprocessing is essential for preparing the dataset for machine learning models. The following steps were implemented:

- **Handling Missing Values**:
  - ” Product_Category_3” had nearly 70% missing values and was dropped due to insufficient data.
  - ” Product_Category_2” had 31.57% missing values, which were imputed using the median to retain consistency without introducing bias.
- **Outlier Removal**: Outliers in the ”Purchase” column were removed using the interquartile range (IQR) method. Data points outside 1.5 times the IQR from the first and third quartiles were excluded to maintain the dataset’s integrity.
- **Feature Encoding**:
  - **Ordered categorical Features**: Attributes such as ”Age” and ”Stay In Current City Years” were encoded using ordinal encoding to preserve their inherent order.
  - **Unordered Categorical Features**: Variables like ”Gender,” ”Occupation,” and ”City Category” were encoded using one-hot encoding. This method generated binary columns for each category, enabling the models to interpret categorical data effectively.

- **Scaling and Normalization**: For neural network models, numerical features were scaled using StandardScaler to normalize the range of values and improve convergence during training.

3.4 Model Implementation

Seven different machine learning models were implemented and evaluated:

- **Linear Regression**: This model served as a baseline to compare the performance of more complex methods. It assumes a linear relationship between the features and the target variable. While simple, its effectiveness is often limited by its inability to capture non-linear relationships.
- **Ridge Regression**: Ridge adds L2 regularization to linear regression to prevent overfitting by penalizing large coefficients. It works well in cases of multicollinearity, where independent variables are highly correlated.
- **Lasso Regression**: Lasso uses L1 regularization to shrink some coefficients to zero, effectively performing feature selection. This is particularly useful in high-dimensional datasets where some features may be irrelevant or redundant.
- **Random Forest**: Random Forest is an ensemble learning method that constructs multiple decision trees and outputs their average prediction. Its ability to capture complex interactions between features makes it a robust model for non-linear data. Additionally, it is less prone to overfitting compared to single decision trees.
- **Gradient Boosting Regressor**: Gradient Boosting trains models sequentially, minimizing residual errors at each step. By focusing on correcting errors made by previous models, it builds a strong predictive model. However, it can be computationally intensive and requires careful tuning to avoid overfitting.
- **XGBoost**: XGBoost (Extreme Gradient Boosting) is a more efficient implementation of gradient boosting with additional features like tree pruning, regularization (L1 and L2), and parallelized computation. Its high flexibility and performance make it a popular choice for structured datasets.
- **Neural Network (MLP)**: Multi-Layer Perceptrons (MLP) are fully connected neural networks capable of modeling complex, non-linear relationships. In this project, an architecture with multiple hidden layers was used, with each layer containing a specific number of neurons and activation functions. The model required extensive tuning of hyperparameters, such as the learning rate, number of layers, and regularization terms, to optimize performance.

4 Experimental Setup

4.1 Training and Testing

- The dataset was split into training (0.80) and testing (0.20) sets to ensure a robust evaluation of model performance. This division allowed for models to be trained on the majority of the data while preserving a separate set for unbiased performance testing.
- For models such as Ridge, Lasso, and Gradient Boosting, k-fold cross-validation was employed to ensure generalization and to prevent overfitting. This method splits the training data into subsets, iteratively training and validating the model on different partitions.
- Numerical features were normalized using StandardScaler for Neural Network models to ensure consistent feature ranges and accelerate convergence during training.
- To reduce dimensionality and improve computational efficiency, irrelevant or redundant features were identified and excluded where necessary.

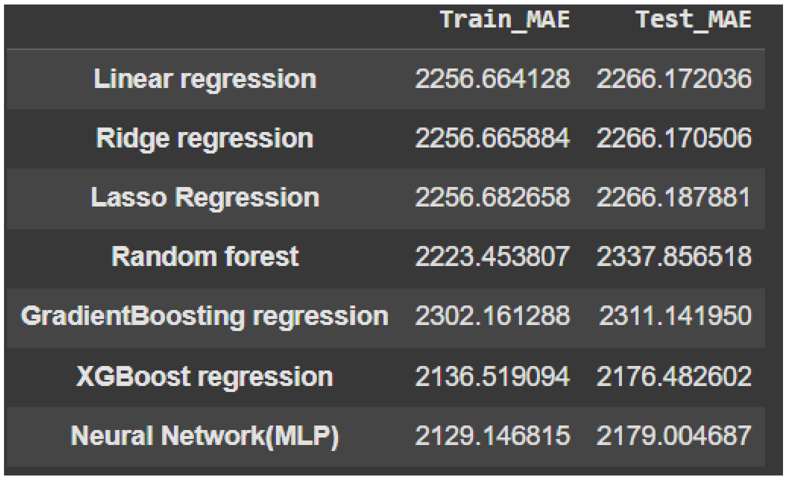
4.2 Hyperparameter Tuning

- **Ridge and lasso**: Optimal regularization parameters (alpha) were determined using GridSearchCV with 5fold cross-validation. This ensured that the chosen hyperparameters minimized error without overfitting the data.
- **Random Forest and Gradient Boosting**: Parameters such as the number of estimators, maximum depth, and minimum samples split were tuned through grid search. These adjustments aimed to balance model complexity and performance.
- **XGBoost**: Multiple hyperparameters, including learning rate, max depth, gamma, and regularization terms (reg lambda and reg alpha), were fine-tuned using grid search. Early stopping was implemented during training to prevent overfitting by monitoring validation loss.
- **Neural Network (MLP)**: Architecture hyperparameters, such as the number of layers, neurons per layer, activation functions, and dropout rates, were adjusted. Optimization techniques, including Adam optimizer and learning rate scheduling, were employed to achieve efficient training.

By employing these strategies, the experimental setup ensured that each model was rigorously optimized and tested under consistent conditions, facilitating a fair comparison of their performances.

5 Result and Analysis

5.1 Model Performance Comparison



	Train_MAE	Test_MAE
Linear regression	2256.664128	2266.172036
Ridge regression	2256.665884	2266.170506
Lasso Regression	2256.682658	2266.187881
Random forest	2223.453807	2337.856518
GradientBoosting regression	2302.161288	2311.141950
XGBoost regression	2136.519094	2176.482602
Neural Network(MLP)	2129.146815	2179.004687

Table 1

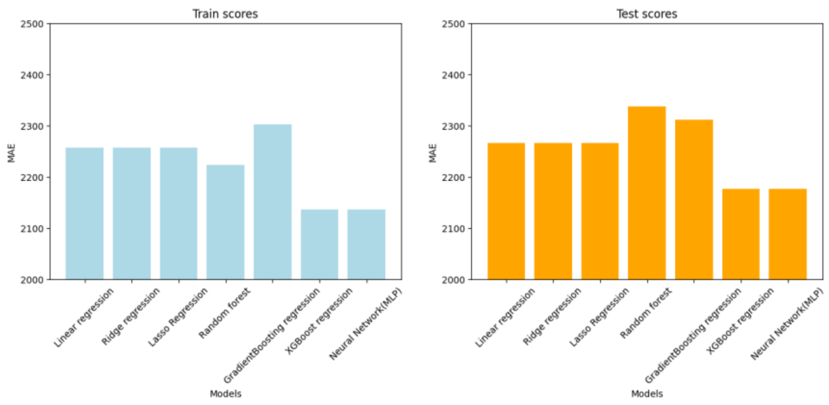


Figure 1 and 2

5.2 insights

- **Linear, Ridge, and Lasso Regression:** When observing that Linear, Ridge, and Lasso regression models yield nearly the same training and testing MAE values, several factors could be at play:
  - **Minimal Regularization Effect:** Ridge and Lasso are regularized versions of linear regression. They aim to reduce model complexity (and possibly improve generalization) by penalizing large coefficients. However, if the chosen regularization parameters (alpha) are very small, the influence of these penalties on the model's coefficient values may be negligible. In such a scenario, Ridge or Lasso will behave almost like a plain Linear Regression model, resulting in nearly identical predictions and error metrics.

  - **Weak Signal-to-Noise Ratio or Model Complexity:** If the relationship between the features and the target is relatively straightforward (i.e., the data is not highly complex or multidimensional in a way that benefits from aggressive regularization), then adding a penalty may not substantially alter the model's coefficients. Lasso, for instance, might be expected to zero out some coefficients if they are not informative. But if all features are genuinely contributing, or if none of them are strongly redundant, the penalty may not have a noticeable pruning effect**.**

  - **Choice of Hyperparameters:** In the provided code, hyperparameter tuning for Ridge and Lasso might have led to an alpha value that is effectively very close to zero or too small to make a meaningful difference. When alpha approaches zero, Ridge and Lasso reduce to standard Linear Regression, because the penalty term fades out.

  - **Data Preprocessing and Scaling**: If the data is already well-conditioned and not prone to overfitting, the benefits of regularization could be minimal. Regularization is most visibly beneficial when the model overfits or when multicollinearity is an issue. If the data is relatively clean and the linear relationship is strong and simple, regularization won't have much room to improve the baseline linear model.

- **Trade-offs:** XGBoost and neural networks (specifically MLPs) can achieve very similar performance for several reasons:
  - **Both Can Model Nonlinearities:** Unlike simple linear models, both XGBoost (a gradient-boosted tree model) and Multi-Layer Perceptrons (fully connected neural networks) can capture complex, nonlinear relationships within the data. If the dataset is well-structured and has strong nonlinear patterns, both models can end up explaining these patterns equally well.

  - **Sufficient Model Complexity:** Both methods are highly flexible and can approximate a wide variety of functions. XGBoost does this by building an ensemble of many weak learners (decision trees), while MLPs do so by learning weights and biases through multiple layers of neurons. If both are given enough capacity (e.g., enough depth or number of trees in XGBoost, and sufficient neurons/layers in MLP) and properly tuned, they might converge to solutions that yield similar error rates.

  - **Effective Regularization and Optimization:** XGBoost incorporates several regularization techniques (like reg_lambda and reg_alpha) and uses gradient boosting to efficiently find patterns. MLPs, when trained with proper parameters (such as suitable regularization, a good learning rate, and a proper initialization), can also settle into effective solutions. Similar levels of well-managed complexity and good hyperparameter tuning can result in comparable performance**.**

– **Plateau in the Accuracy vs. Complexity Curve:** On some datasets, especially those where adding complexity yields diminishing returns, multiple sophisticated models can reach a performance "plateau." In such a scenario, once the data's underlying patterns are captured sufficiently well, adding complexity or switching to a different flexible model does not lead to drastically different performance. Both XGBoost and MLP might be hitting this plateau, thus resulting in similar MAE values.

In short, XGBoost and MLPs are both powerful, flexible modeling methods. When both are well-tuned and the dataset is not favouring one approach distinctly over the other, it's not uncommon for their performance metrics to be very close

5.3 Feature Importance

The XGBoost model revealed the following as key predictors of sales:

- Product category features
- Customer demographics (e.g., age, city category)

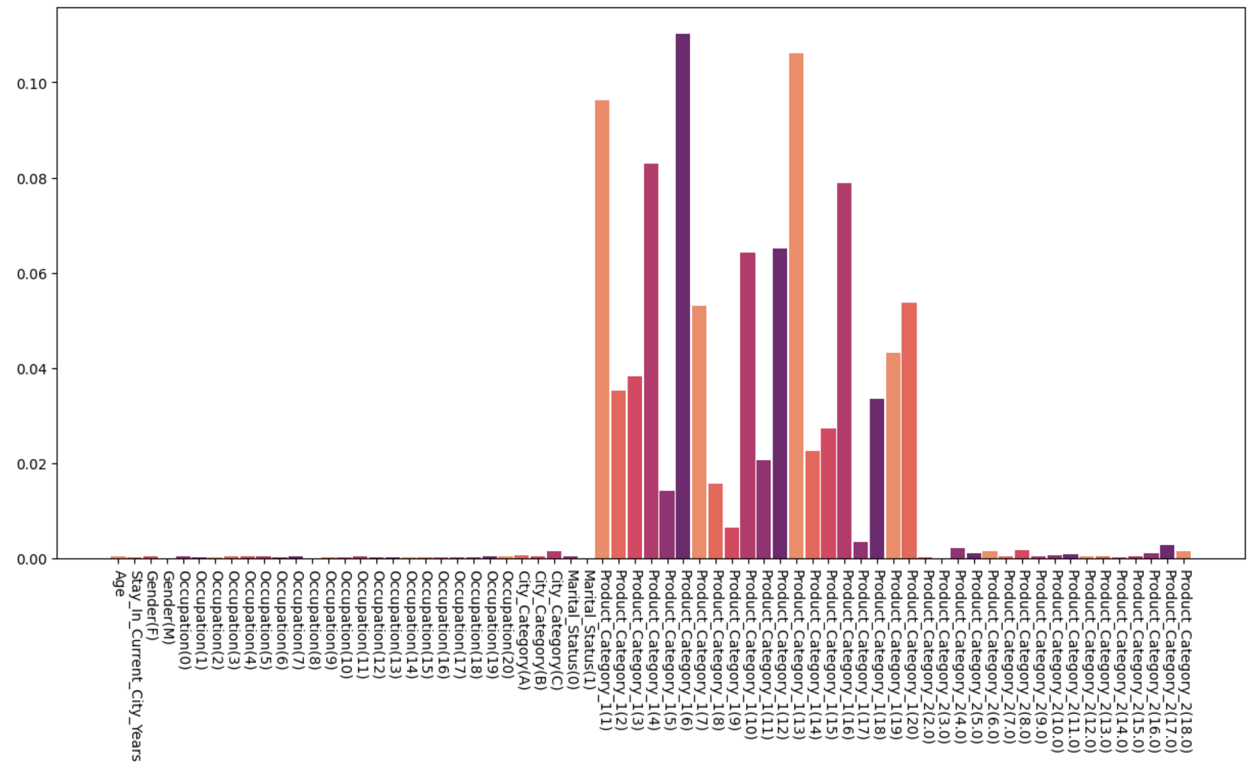


Figure 3

6 Discussion

6.1 Practical Implications

- **Retail Planning:** By identifying the key drivers of sales, such as product categories and demographic factors, retailers can optimize their inventory management strategies. For instance, stores in specific city categories might stock more of the products predicted to perform well during high-sales periods like Black Friday.
- **Dynamic Pricing:** The ability to predict purchase amounts with high accuracy enables retailers to implement dynamic pricing strategies. Using real-time sales data and model predictions, businesses can adjust prices to maximize revenue and respond to market demand efficiently**.**
- **Targeted Marketing**: Insights into demographic patterns, such as age groups or city categories, allow for more personalized marketing campaigns. Retailers can allocate promotional budgets to target the most profitable customer segments effectively, thereby improving return on investment.
- **Strategic Resource Allocation:** Sales forecasting models can inform staffing and logistics during peak sales events, ensuring resources are allocated where they are needed most.

6.2 Limitations

- **Dataset-Specific Challenges:** The dataset focuses exclusively on Black Friday, a unique sales event. As a result, the model's insights and predictions may not generalize to other time periods or sales scenarios**.**
- **Model Complexity and Resource Requirements:** Advanced models such as Neural Networks require substantial computational power and extensive hyperparameter tuning, which may not be feasible for smaller businesses.
- **Potential Overfitting:** Despite careful tuning, some models may still overfit to the training data, reducing their ability to perform consistently on unseen datasets.
- **Feature Constraints:** The dataset lacks certain potentially influential features, such as competitor pricing or external economic factors, which could improve model accuracy if included.

6.3 Future Directions

- **Incorporating External Data:** Future work could enhance model performance by including external variables, such as economic indicators, weather patterns, or competitor pricing data.
- **Extending Beyond Black Friday**: Expanding the dataset to include sales data from other periods would enable the development of more generalized forecasting models.
- **Exploring Advanced Architectures:** Further research could explore architectures like Transformer models or convolutional neural networks (CNNs) for more robust pattern recognition.
- **Real Time Forecasting:** Implementing models in real-time retail systems could further optimize decision-making processes, enabling on-the-fly adjustments to inventory, pricing, and promotions.

7 Conclusion

This project demonstrated the potential of machine learning models in sales forecasting, with XGBoost and Neural Networks emerging as the most effective approaches. The findings highlight the importance of feature engineering and hyperparameter tuning in achieving optimal performance. Future work could extend this approach to other retail datasets and explore additional deep learning architectures.

8 References

- Kaggle Black Friday Sales Dataset
- Breiman, L. (2001). Random Forests. Machine Learning
- Yue Ning. In-class lecture presentations, CS 559, Stevens Institute of Technology.
- Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine.

9 Appendix

- **Visualizations**
  - Model MAE Comparison
    * Train Scores: See Figure 1
    * Test Scores: See Figure 2
  - Feature Importance from XGBoost: See Figure 3

- **Dataset Summary**

  Key attributes and their distributions are documented in the exploratory data analysis section of the notebook.