Project Report

Implementing and Evaluating Custom Machine Learning Models on Sales Prediction

Kabir Uberoi[1] and Shikhar Gupta[2]

[1]Department of Mathematics,Indian Institue Of Technology, Delhi

November 29, 2024

Abstract

Accurate sales forecasting plays a pivotal role in optimizing inventory management, formulating effective pricing strategies, and driving profitability within the retail sector. This study examines the application of machine learning models, implemented from first principles, for sales prediction and evaluates their performance against established models provided by scikit-learn. Leveraging a dataset encompassing product and store features, the models are assessed using Mean Squared Error (MSE) and R^2 metrics. The findings offer valuable insights into the viability and effectiveness of custom implementations in predictive analytics. The source codes can be seen at:
https://github.com/KabirUberoi/Sales-Prediction-Model

Keywords: sales prediction, machine learning, regression models, scikit-learn, custom implementation.

1 Introduction

Accurate sales prediction is a critical challenge in retail analytics. This project focuses on developing predictive models to estimate product sales (Item Outlet Sales) using a combination of product and store features. The dataset comprises 8,000 records with attributes such as product weight, visibility, category, outlet type, and maximum retail price.

The task is framed as a regression problem, with the objective of minimizing the Root Mean Squared Error (RMSE) between predicted and actual sales values. Robust prediction models have the potential to empower businesses to make data-driven decisions, streamline operations, and enhance customer satisfaction.

The study involves:

- Preprocessing the dataset and feature engineering.
- Evaluating regression models using scikit-learn to benchmark performance.
- Implementing selected models from scratch to explore their theoretical underpinnings and practical feasibility.

2 Data and Methods

2.1 Dataset Description

The dataset includes the following features:

- **Product Features:** *Item_Weight, Item_Fat_Content, Item_Visibility, Item_Type, Item_MRP.*
- **Store Features:** *Outlet_Identifier, Outlet_Establishment_Year, Outlet_Size, Outlet_Location_Type, Outlet_Type.*
- **Target Variable:** *Item_Outlet_Sales*, representing the sales figures for products in specific outlets.

2.2 Models Evaluated with Scikit-learn

We initially tested various regression models using the scikit-learn library. The models and their performance, measured using MSE and R^2 scores, are summarized in Table 1.

2.3 Custom Implementations

Based on the scikit-learn results, the following models were selected for custom implementation:

1. Gradient Boosting
2. Polynomial Regression
3. Random Forest
4. Linear Regression
5. Ridge Regression

Each model was built from scratch, adhering to its theoretical principles, and evaluated on the dataset.

Table 1: Performance of scikit-learn models.

Model	MSE	R^2
MLP Regressor	1.172×10^6	0.604
Gradient Boosting	1.176×10^6	0.603
Polynomial Regression	1.209×10^6	0.592
Linear Regression	1.315×10^6	0.556
Ridge Regression	1.315×10^6	0.556
Lasso Regression	1.315×10^6	0.556
SGD Regressor	1.324×10^6	0.553
Bagging Regressor	1.363×10^6	0.540
XGBoost Regressor	1.393×10^6	0.529
KNeighbors Regressor	1.497×10^6	0.494
AdaBoost Regressor	1.521×10^6	0.486
Random Forest	1.627×10^6	0.479
Support Vector Regressor (SVR)	2.776×10^6	0.062

3 Results

The performance of the custom implementations is summarized in Table 2.
The optimal hyperparameters for all models were determined through extensive hyperparameter tuning. The results are as follows:

Random Forest:

- Max Depth: 10
- Number of Trees: 155
- Minimum Samples Split: 2
- Number of Features: 36

Gradient Boosting:

- Number of Trees: 50
- Max Depth: 3
- Minimum Samples Split: 2
- Learning Rate: 0.1

Regression Models:

- **Linear Regression:**
  - Learning Rate: 10^{-4}
  - Number of Epochs: 1000
- **Ridge Regression:**
  - Learning Rate: 10^{-3}
  - Number of Epochs: 500
- **Polynomial Regression:**
  - Learning Rate: 10^{-4}
  - Number of Epochs: 1000
  - Degree of Polynomial: 2

Table 2: Performance of custom-implemented models.

Model	MSE (Custom)
Gradient Boosting	1.3735e6
Polynomial Regression	1.2709e6
Random Forest	1.9088e6
Linear Regression	1.2713e6
Ridge Regression	1.3320e6

4 Conclusion

We conclude that regression models outperformed the Decision Tree-based models. Among the Decision Tree-based models, Gradient Boosting achieved better performance compared to Random Forest.

References

[1] Lectures of ELL-409. https://lcs2-iitd.github.io/ELL409-2401/lectures/

[2] Scikit-learn Documentation. https://scikit-learn.org/

[3] "Random Forests" by Leo Breiman (2001). https://link.springer.com/article/10.1023/A:1010933404324

[4] "Greedy Function Approximation: A Gradient Boosting Machine" by Jerome H. Friedman (2001)https://link.springer.com/article/10.1023/A:1010933404324

[5] "Ridge Regression: Biased Estimation for Nonorthogonal Problems" by Arthur E. Hoerl and Robert W. Kennard(2012). https://www.tandfonline.com/doi/abs/10.1080/00401706.1970.10488634