This assignment showcases a regression approach to predictive analytics in inventory Management for Amway.

Optuna-Tuned XGBoost for Inventory Management - The Amway Case Study

Optimising Inventory Management through Predictive Analytics for Competitive Advantage

Name: Mubanga Nsofu

Learner ID: 149050

Module: BAN6440 (Applied ML for Business Analytics)

Lecturer: Prof. Bucciarelli

Date: 16th November 2024

Module 2 Assignment

**1.0 Introduction:**

This assignment uses supervised learning to focus on effective inventory management for the fast-moving consumer goods (FMCG) market. To this end, XGBoost (Chen & Guestrin, 2016), a supervised learning algorithm, is used on synthetic data to perform a regression task. Ensemble models have gained traction over the last several years—for instance, the Random Forest algorithm developed by Ho (1995). Ensemble approaches are powerful models as many weak learners can be combined to improve overall performance (Rocca, 2019). This approach has been chosen to optimise inventory management in the FMCG sector and enable Amway to make better decisions.

**2.0 Synthetic Data Generation:**

A synthetic dataset of 300 records for the Amway inventory is created considering the following features:

* Product ID (Unique Identifier)
* Monthly Sales Volumes (Units sold)
* Inventory Levels (Current Stock)
* Reorder Point (Threshold level for reordering)
* Lead Time (In days)
* Supplier Reliability Score (1 – 10 scale)
* Price per unit (USD)
* Daily Stock Out (Predicted days until inventory depletion)

Daily Stock Out is the target variable, and regression will be used to solve the prediction problem.

The code in Figure 1 below generates the synthetic data.

Figure 1

Synthetic data generation in Python.



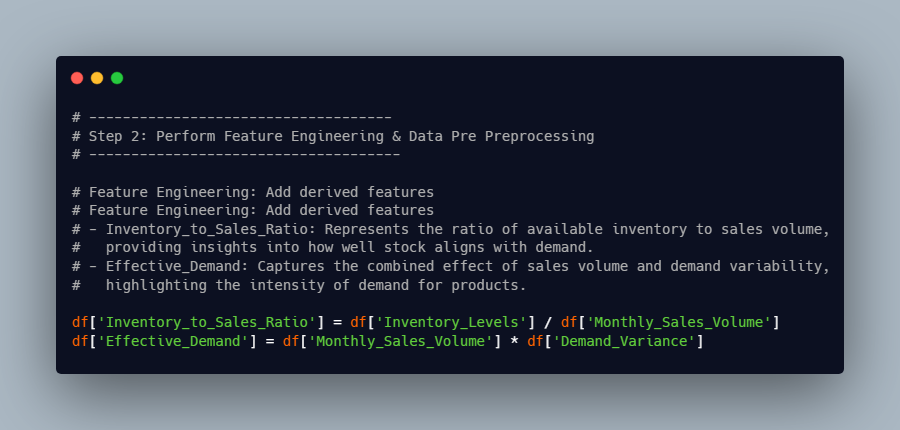
Note. Created by Mubanga Nsofu, November 2024.

**3.0 Feature Engineering and Data Preprocessing:**

The data is synthetic, and the code above does not provide for missing values. In a real-world dataset, a data engineer would need to deal with missing values and other data quality issues that may arise before proceeding with the modelling. However, using feature engineering, derived features are added. The derived features ensure the model can reach its full potential by capturing interactions not fully represented in the current dataset’s features. This is shown in Figure 2 below.

Figure 2:

Feature Engineering to introduce derived features



Note. Created by Mubanga Nsofu, November 2024.

After feature engineering, data preprocessing is essential. This ensures the dataset is compatible with the machine learning algorithm, in this case, XGBoost. In the code snippet below, the following steps are carried out:

* Application of one hot encoding: needed to deal with categorical features in the dataset; Product Category is converted into multiple binary columns
* Exclude the first category to avoid redundancy
* Define the target variable
* Split the dataset into a training and test set using a 70:30 ratio. The latter is used to evaluate the model's performance.

Figure 3

Data Preprocessing



Note. Created by Mubanga Nsofu, November 2024

**4.0 Instantiate XGBoost Model & Tune Hyperparameters:**

According to Nvidia (2024), Extreme Gradient Boosting (XGBoost) is an algorithm of high accuracy and computational efficiency. XGBoost builds upon previous algorithms such as decision trees, random forests, and gradient-boosting decision trees. XGBoost was chosen for this problem as it has regularisation to prevent overfitting, incorporates tree pruning to limit the individual tree complexity, and, most importantly, parallel processing to speed up training times. The XGBoost Regressor model applied to the Amway case study is shown in the figure below within a function designed to tune the hyperparameters using Optuna.

Figure 4

Baseline XGBRegressor

A screenshot of a computer program

Description automatically generated

Note. Created by Mubanga Nsofu, November 2024.

Figure 5

objective function containing the initialized XGBRegressor

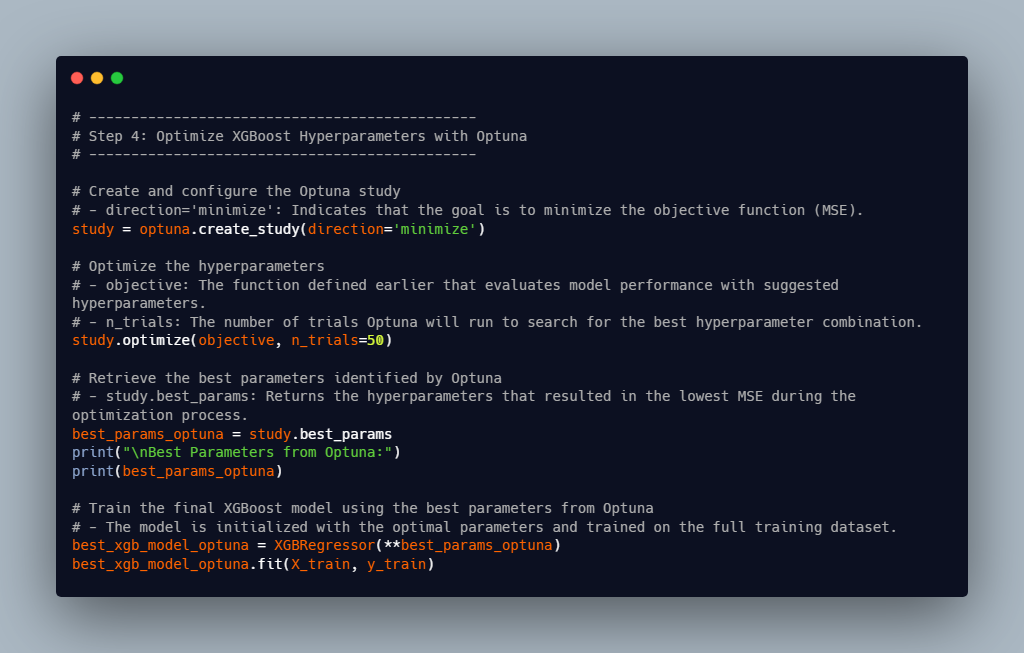


Note. Created by Mubanga Nsofu, November 2024.

The following code snippet optimises the hyperparameters with Optuna. Hyperparameter tuning improves a model's performance. In this use case, finding the optimal parameters reduced the overall loss.

Figure 6

Hyperparameter tuning



Note. Created by Mubanga Nsofu, November 2024.

**5.0 Baseline XGBoost versus Optimised XGBoost Model Evaluation and Interpretation:**

The models were tested on the test dataset. The following metrics were used to evaluate the regression model.

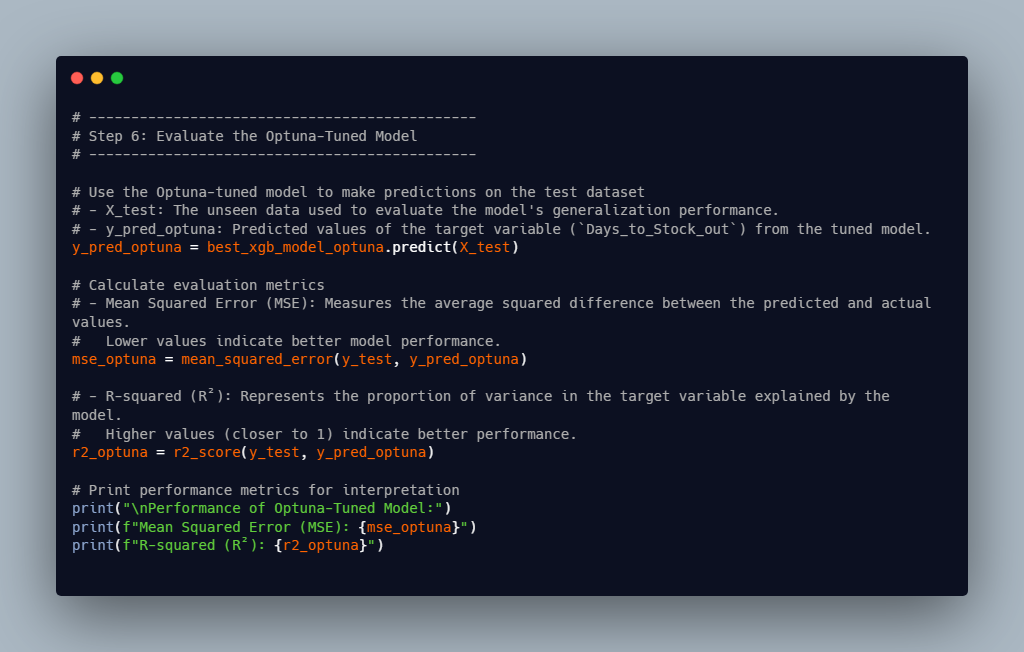
|  |  |  |
| --- | --- | --- |
| Metric | Baseline XGBoost | Optimised XGBoost |
| MSE | 0.3956 | 0.1560 |
| R2 | 61.45% | 84.42% |

**Mean Squared Error (MSE):** This is the sum of the squared residuals. The MSE is used during the optimisation process to minimise the prediction errors. After tuning, the model's output was Mean Squared Error (MSE): 0.1560 compared to 0.3956.The average squared difference between predicted and actual values is now minimal for the tuned XGBoost model, showing that the model predicts Days\_to\_Stock\_out with high accuracy.

* R-squared (R2): This is the coefficient of determination. It is a metric that explains how much variance the model explains. After tuning, the output from the model was R²: 0.8442 compared to the baseline of 0.6145. The optimised model explains 84.42% (a significant proportion) of the variance in the target variable (Days\_to\_Stock\_out). The code snippet below shows how the model was evaluated.

Figure 7

XGBoost Model evaluation



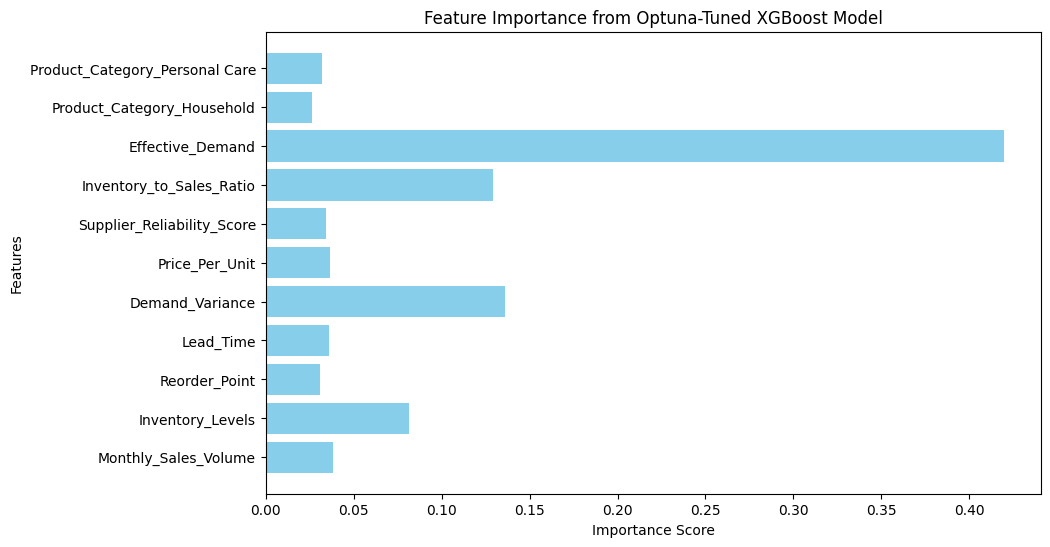
Note. Created by Mubanga Nsofu, November 2024.

**6.0 Visualization of Feature Importance:**

The chart below shows the importance of each feature after XGBoost hyperparameter tuning. Effective Demand, inventory-to-sales ratio, and Demand variance have the highest importance scores.

Figure 8

Feature importance from Optuna-tuned XGboost Model



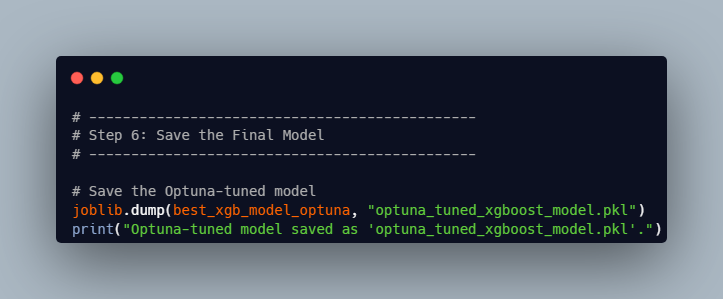
Note. Created by Mubanga Nsofu, November 2024

**7.0 Save The final model:**

The tuned model is saved using the code below.

Figure 9

Code snippet to save the model.



Note. Created by Mubanga Nsofu, November 2024

**8.0 Conclusion:**

XGBoost was applied to a synthetic dataset to demonstrate the process of preparing and preprocessing data, modelling, tuning hyperparameters, and evaluating performance using a test set. The model explains 84.42% (a significant proportion) of the variance in the target variable (Days\_to\_Stock\_out). XGBoost effectively captures the complex non-linear aspects of the dataset.

The average squared difference between predicted and actual values is minimal (MSE of 0.1560), showing that the model predicts Days\_to\_Stock\_out with high accuracy.

Using an XGBoost regressor, Amway can better manage its stock inventory and make smarter decisions.

**9.0 References:**

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13*-*17*-*August*-*2016*, 785–794. <https://doi.org/10.1145/2939672.2939785>

Nvidia. (2024). *XGBoost – What Is It and Why Does It Matter?* https://www.nvidia.com/en-us/glossary/xgboost/

Tin Kam Ho. (1995). Random decision forests. *Proceedings of 3rd International Conference on Document Analysis and Recognition*, *1*, 278–282. https://doi.org/10.1109/ICDAR.1995.598994

Rocca, J. (2019). *Ensemble methods: bagging, boosting and stacking – Towards Data Science*. https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205