Research & Development Document

Title: Unit Sales Forecasting without Ad Spend Data

1. Introduction

The objective of Task 2 in this Kaggle competition is to predict unit sales without relying on ad spend data. This document provides a comprehensive overview of the methods and techniques used to achieve accurate sales predictions, focusing on feature engineering, model selection, and evaluation.

2. Objective

The primary goal is to develop a predictive model that can accurately forecast unit sales using historical sales data and other relevant features, while excluding ad spend data. This is crucial for scenarios where ad spend information is unavailable or unreliable.

3. Data Understanding and Preparation

3.1 Dataset Overview

The dataset consists of two files: train.csv and test.csv. The training set includes the following columns:

- ID
- date
- Item Id
- Item Name
- ad_spend
- anarix_id
- units
- unit_price

The test set includes all columns except for units, which is the target variable.

3.2 Data Cleaning and Preprocessing

To ensure data quality and consistency, the following preprocessing steps Ire undertaken:

- **Date Conversion**: The date column was converted to a datetime format to facilitate time-series analysis.
- **Handling Missing Values**: Rows with missing values generated by lag features Ire dropped to maintain data integrity.

4. Feature Engineering

4.1 Lag Features and Rolling Statistics

Lag features and rolling statistics Ire created to capture temporal dependencies in the sales data. These features help the model understand trends and patterns over time.

- **Lag Features**: Created lagged versions of the units column (from 1 to 7 days).
- **Rolling Mean and Standard Deviation**: Calculated the rolling mean and standard deviation over a 7-day window.

These features Ire essential in providing the model with historical context, allowing it to learn from past sales data.

5. Model Selection and Training

5.1 Choice of Model: XGBoost

XGBoost was selected as the predictive model due to its robust performance in regression tasks and its ability to handle complex, high-dimensional data. XGBoost's gradient boosting framework combines multiple Iak learners to form a strong predictive model.

5.2 Data Encoding

Categorical variables, such as Item Id and anarix_id, Ire label encoded to convert them into numerical representations. This step was crucial for the model to process categorical data effectively.

6. Training Process

The training process involved splitting the dataset into training and validation sets to evaluate the model's performance. Key steps included:

- **Label Encoding**: Transformed categorical features into numerical values.
- **Model Training**: Trained the XGBoost model using the training set.
- **Evaluation**: Assessed the model's performance on the validation set using Mean Squared Error (MSE).

7. Generating Predictions

The trained model was used to generate predictions for the test set. Since the test set lacked the units column, lag features and rolling statistics Ire created based on the last available data from the training set.

8. Results and Evaluation

The model's performance was evaluated on the validation set, achieving a satisfactory MSE. The predictions Ire formatted according to the competition's requirements and saved for submission.

9. Conclusion

This R&D document details the end-to-end process of forecasting unit sales without ad spend data. The combination of feature engineering, robust model selection (XGBoost), and careful data preprocessing contributed to the success of the predictive model.

10. Future Work

To further improve model performance, future work could explore:

- **Advanced Time-Series Features**: Incorporating more sophisticated time-series features.
- **Alternative Models**: Experimenting with different machine learning models, such as recurrent neural networks (RNNs) or Long Short-Term Memory (LSTM) networks.
- **Hyperparameter Tuning**: Conducting extensive hyperparameter tuning to optimize model performance.

References

- Kaggle competition details and dataset
- XGBoost documentation
- Scikit-learn documentation for data preprocessing techniques