Merchandise Sales and Profits Predictions, and Image Classification

Full Name: Mengzhe (Alvin) Zhao

Email: mzha910@aucklanduni.ac.nz

Analysis and Preprocessing - Merchandise Sales and Profits Predictions

Data Understanding

W Store

1. Features Dataset: ['Store', 'Date', 'Temperature', 'Fuel_Price', 'MarkDown1', 'MarkDown2', 'MarkDown3', 'MarkDown4', 'MarkDown5', 'CPI', 'Unemployment', 'IsHoliday']
2. Sales Dataset: ['Store', 'Dept', 'Date', 'Weekly_Sales', 'IsHoliday']
3. Stores Dataset: ['Store', 'Type', 'Size']
4. Combined Dataset: ['Store', 'Date', 'Temperature', 'Fuel_Price', 'MarkDown1', 'MarkDown2', 'MarkDown3', 'MarkDown4', 'MarkDown5', 'CPI', 'Unemployment', 'IsHoliday', 'Weekly_Sales', 'Type', 'Size']

X Store

1. Sales Dataset: ['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode', 'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'State', 'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-Category', 'Product Name', 'Sales', 'Quantity', 'Discount', 'Profit', 'Order Year', 'Order Month', 'Order Day', 'Ship Year', 'Ship Month', 'Ship Day']

Exploratory Data Analysis (EDA)

W Store

1. There are obvious differences between different stores for CPI distribution.
2. There is a seasonal trend for temperature time trend.
3. The data of features and label have several types of data distribution.
4. Holidays have lower temperature and lower fuel price.
5. The descending order of store number and size is A, B, and C.

X Store

1. The numeric features have several types of data distribution.
2. There is a seasonal trend for the number of orders.
3. Chairs have the highest sales and profit.

4. New York is the city with the highest sales, Seattle is the city with the highest profit.
5. California is the state with the highest sales and profit.

Data Cleaning

1. For W stores, I combine three datasets into one whole dataset based on 'Store' and 'Date'.
2. For missing values, only W store dataset has them, X store has no.
3. For outliers, I use IQR to detect them.
4. For feature engineering, I extract year, month, and day of each date as new features.
5. I establish a pipeline to work for data preprocessing, which contains imputing missing values using median, scaling data using Robust Scaler, and converting categorical features to numeric features using one-hot encoding. Because Robust Scaler can reduce the influence of outliers, I do not use homemade outlier handler.

Corelated Variables

1. For W stores, the size of each store has the most noticeable correlation between the weekly sales.
2. For X stores, the sales of each order have the most noticeable correlation between the profit.

Training and Evaluation - Merchandise Sales and Profits Predictions

What I have found during training and evaluation

1. For W store and X store, Gradient Boosting always perform better than Lasso Regression.
2. For both regression models, W store always perform better than X store. The reason for this may be W store has much more data than X store.

Training and evaluation steps

Lasso Regression

1. Model initialized with alpha=1.0.
2. Model trained on the training dataset.
3. Predictions made on both training and test datasets.

4. Errors calculated: MAE (71.81), MSE (19168.15), RMSE (138.45), R2 (0.16), SMAPE (122.25), and MAPE (inf.).
5. Metrics printed for both training and test sets.

Gradient Boosting

1. Model initialized with 100 estimators, learning rate of 0.1, and max depth of 3.
2. Model trained on the training dataset.
3. Predictions made on both training and test datasets.
4. Errors calculated: MAE (41.58), MSE (11185.38), RMSE (105.76), R2 (0.51), SMAPE (82.04), and MAPE (inf.).
5. Metrics printed for both training and test sets.

Next Steps to Improve the Models

Lasso Regression

1. Hyperparameter Tuning: optimise the regularisation parameter ('alpha') using cross-validation.
2. Feature selection: remove irrelevant or highly correlated features to reduce overfitting.

Gradient Boosting

1. Hyperparameter Tuning: optimise parameters such as ‘n_estimators', 'learning_rate', 'max_depth', 'and min_samples_split' using grid search.
2. Early stopping: use early stopping to prevent overfitting by monitoring validation loss.
3. Boosting variants: Experiment with different boositng algorithms like XGboost for potentially better performance.

Deep Learning – Image Classification

What I have found during training and evaluation

1. For this image classification problem with lots of data, CNN model (0.61 accuracy) always performs better than MLP model (0.09 accuracy), which corresponds with the usage cases for these two deep learning algorithms.

Training and evaluation steps

MLP model

1. Data reshaped to a 2D format.
2. MLP model built with dense layers and dropout for regularization.
3. Model trained for 20 epochs with a batch size of 64.
4. Learning rate schedule applied.
5. Initial evaluation on validation data during training.
6. Model retrained using combined training and validation sets, and predictions made on test data.

CNN model

1. Data reshaped to a 4D format suitable for CNNs.
2. CNN model built with convolutional layers, batch normalization, and dropout.
3. Model trained for 20 epochs with a batch size of 64 using augmented data.
4. Learning rate schedule applied.
5. Initial evaluation on validation data during training.
6. Model retrained using combined training and validation sets, and predictions made on test data.

Next steps to improve the models

MLP model

1. Tune hyperparameters.
2. Experiment with different activation functions.
3. Increase training epochs and/or batch size.
4. Implement early stopping to avoid overfitting.

CNN model

1. Tune hyperparameters.
2. Experiment with different activation functions.
3. Increase training epochs and/or batch size.
4. Implement early stopping to avoid overfitting.