Anuththara Divyanjalie

Machine Learning Internship/Associate Assignment

Table of Contents

1. Executive Summary 4

2. Tasks Overview 5

2.1 Project Context....5

2.1.1 Background of the Supermarket Data Analysis 5

2.1.2 Timeline and Scope 5

2.1.3 Available Datasets and Their Relationships....5

2.2 Challenges and Approach....6

2.2.1 Key Challenges Encountered 6

2.3 Methodology 8

2.4 Tools and Technologies Used 8

3. Data Preprocessing....9

3.1 Preprocessing Steps....9

3.1.1 General Data Cleaning Methodology....9

3.2 Dataset-Specific Preprocessing Steps 12

3.2.1 Items Dataset 12

3.2.2 Sales Dataset....14

3.2.3 Promotions Dataset....15

3.2.4 Supermarkets Dataset 15

3.3 Merging Process: Integration of Dataset....16

3.3.1 Merging Strategy 16

4. Supervised Learning Model (Task 01) 17

4.1 Problem One....17

4.1.1 Business Problem Statement One 17

4.1.2 Objectives and Success Criteria....17

4.1.3 Expected Business Impact 17

4.1.4 Model Development....18

4.1.5 Model Evaluation....19

4.1.6 Results Analysis....20

4.1.7 Model Limitations....21

4.1.8 Business Insights....21

4.1.9 Business Recommendations....21

4.1.10 Potential Impact and Value 21

4.1.11 Implementation Considerations....22

4.2 Problem Two 22
4.2.1 Business Problem Statement 22
4.2.2 Objectives and Success Criteria 22
4.2.3 Expected Business Impact 23
4.2.4 Model Development 23
4.2.5 Model Evaluation 24
4.2.6 Results Analysis 25
4.2.7 Model Limitations 26
4.2.8 Validation of Results 27
4.2.9 Business Insights 27
4.2.10 Business Recommendations 27
4.2.11 Implementation Considerations 28
5. Maze Navigation Model (Task 02) 29
5.1 Environment Design 29
5.2 Reward System Design 30
5.3 Reinforcement Learning Implementation 30
5.4 Performance Analysis 31
5.6 Technical Implementation 33
5.6.1 Code Structure 33
6. Conclusion 34

1. Executive Summary

This report outlines the analysis and findings from a project focused on supermarket transaction data collected over a two-year period. The objective of the project is to harness machine learning techniques to derive actionable business insights from the provided datasets, which include detailed information on items for sale, sales transactions, promotions, and supermarket locations. Additionally, the project incorporates a reinforcement learning (RL) task aimed at developing a model that learns to navigate a maze environment. This RL model optimizes its path through trial-and-error interactions, simulating the complexities of dynamic and uncertain environments, further demonstrating the application of machine learning techniques in both predictive analytics and real-time decision-making.

The project is divided into two main tasks:

Supervised Learning Model Development	The first task involves cleaning, normalizing, and transforming the datasets into Python-compatible formats, followed by the implementation of a supervised learning model. Problem statements are defined to address specific issues. A suitable supervised learning algorithm is employed to develop the model. The model is evaluated using key performance metrics such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and accuracy. This evaluation yields insights that can drive strategic business decisions, such as optimizing inventory management and enhancing promotional strategies.
Reinforcement Learning for Maze Navigation	The second task centers around designing a reinforcement learning model capable of navigating a maze. This model learns to optimize its path through trial-and-error interactions within a simulated environment. By receiving feedback based on its actions, the model progressively improves its navigation capabilities, demonstrating the potential of reinforcement learning in dynamic and uncertain contexts.

This report documents the methodology and processes undertaken for both tasks, including data cleaning, feature selection, model training, and evaluation. It provides a detailed analysis of the business insights generated from the supervised learning model, highlighting their relevance and applicability to real-world supermarket operations. An overview of the maze navigation model is presented, discussing the reinforcement learning techniques employed and the performance outcomes observed during training.

The findings of this project illustrate the power of machine learning in extracting valuable insights from complex datasets and demonstrate the potential for developing intelligent systems capable of solving dynamic problems. The results serve as a foundation for further exploration and enhancement of machine learning applications.

2. Tasks Overview

2.1 Project Context

2.1.1 Background of the Supermarket Data Analysis

This project involves analyzing transaction data from multiple supermarket branches collected over a two-year period. The supermarkets are located across two main provinces and offer four distinct item types (Type 1 to Type 4). The primary goal is to derive actionable business insights using machine learning techniques while also exploring reinforcement learning applications.

2.1.2 Timeline and Scope

- **Project Duration:** One calendar week
- **Scope includes:**
  - Data preparation and cleaning of supermarket transaction datasets.
  - Development of supervised learning models to generate at least two business insights.
  - Implementation of a reinforcement learning model for maze navigation.
  - Documentation and reporting of findings.

2.1.3 Available Datasets and Their Relationships

Items.csv	• Contains product information including code, description, type, brand, and size.

	• Serves as the master product catalog.
Sales.csv	• Two years of transaction data. • Key fields: code, amount, units, transaction time, province, customer ID, supermarket number, basket, day, voucher
Promotion.csv	• Promotional campaign details. • Contains item code, supermarket number, week, feature, display, and province.
Supermarkets.csv	• Store location details including supermarket number and post-code. • Forms the base reference for store-level analysis.

Relatioships

- **Item.csv and Sales.csv are linked through the "Code" field:**
- One item (identified by Code) can appear in multiple sales records.
- This appears to be a one-to-many relationship from Item to Sales.
- **Sales.csv and Supermarket.csv are connected via "Supermarket No":**
- One supermarket can have multiple sales records.
- This is a one-to-many relationship from Supermarket to Sales.
- **Promotion.csv and Item.csv are connected through the "Code" field:**
- One item can have multiple promotions.
- This appears to be a one-to-many relationship from Item to Promotion.
- **Promotion.csv and Supermarket.csv are linked via "Supermarkets":**
- This suggests promotions can be applied to specific supermarkets.
- The relationship appears to track which promotions are running at which stores.

2.2 Challenges and Approach

2.2.1 Key Challenges Encountered

Data Integration	Merging multiple datasets while maintaining data integrity. Handling relationships between different data sources. Ensuring consistency across time periods.

Data Quality	Identifying and handling missing values. Detecting and addressing outliers. Normalizing data across different stores and provinces.
Supervised Learning Model Development	Feature selection for meaningful business insights. Balancing model complexity with interpretability Ensuring model generalization across different store locations.
Reinforcement Learning Implementation	Designing an appropriate maze environment with realistic constraints. Defining effective reward structures to encourage optimal path finding. Balancing exploration vs. exploitation in the learning process. Managing computational resources during training iterations. Ensuring the agent can generalize to different maze configurations.
Technical Integration	Managing computational resources across both tasks. Developing consistent evaluation metrics for both models. Creating scalable solutions that can handle increasing complexity. Implementing proper documentation for both systems.
Performance Optimization	Fine-tuning hyperparameters for both supervised and reinforcement learning models. Optimizing training time while maintaining model quality. Developing efficient evaluation pipelines for both tasks. Ensuring real-time performance for the maze navigation system.

2.3 Methodology

1. **Data Preparation Phase**
   - Initial data exploration and quality assessment.
   - Data cleaning and normalization.
   - Feature engineering and selection.
2. **Supervised Learning Implementation**
   - Problem definition and model selection.
   - Model training and validation.
   - Performance evaluation and optimization.
3. **Reinforcement Learning Development**
   - Maze environment design.
   - Agent implementation.
   - Training and optimization.

2.4 Tools and Technologies Used

1. **Programming Language**
   - Python 3.x for all development work.
2. **Data Processing Libraries**
   - Pandas for data manipulation and analysis.
   - Numpy for numerical computations and array operations.
   - Tabulate for formatted table display.
   - Scipy.stats for statistical operations.
3. **Machine Learning Libraries**
   - Scikit-learn for:
     - Data preprocessing (LabelEncoder, OneHotEncoder).
     - Model implementation (RandomForestRegressor, RandomForestClassifier).
     - Model evaluation (metrics like MSE, R2, MAE).
     - Hyperparameter tuning (RandomizedSearchCV).
4. **Visualization Tools**
   - Matplotlib for static plots and visualizations.
   - Seaborn for enhanced statistical visualizations.

   - Streamlit for interactive maze simulation interface.

5. **Development Environments**
   - Kaggle Jupyter Notebooks for model development and analysis.
   - Local development environment for Streamlit application.
6. **Additional Libraries**
   - Collections (deque) for efficient list operations in RL.
   - Pickle for model serialization.
   - Re for regular expression operations.
   - Fractions for numerical computations.
7. **Version Control and Project Management**
   - Git for code versioning and collaboration.
8. **Deployment Tools**
   - Streamlit for creating and deploying the interactive maze simulation.

3. Data Preprocessing

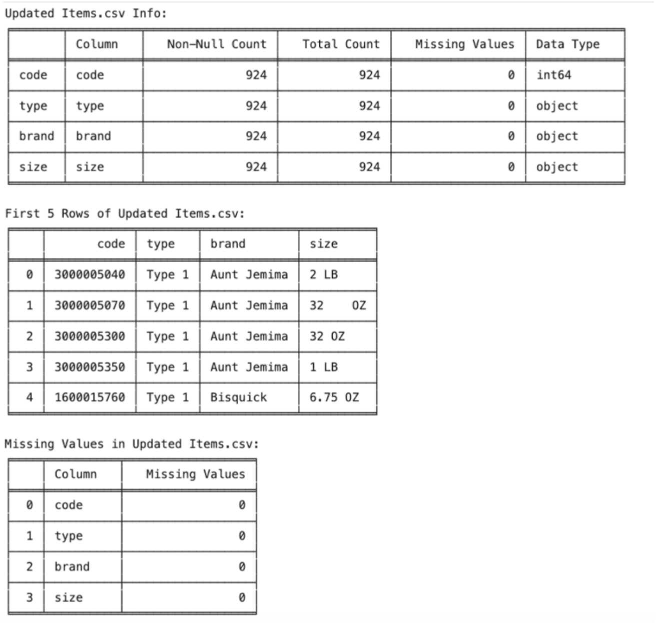
3.1 Preprocessing Steps

Data preprocessing is a crucial step in the data analysis pipeline that ensures datasets are clean, consistent, and ready for analysis. This section outlines the preprocessing steps applied to the items, sales, promotions, and supermarket datasets to prepare them for meaningful insights and analytical tasks.

3.1.1 General Data Cleaning Methodology

1. Handling Missing Values:

- Common empty representations (e.g., empty strings) were replaced with *NaN* using the *replace* method to standardize missing data representation.
- Summary tables were generated for each dataset, identifying missing values per column to assess data quality and inform cleaning actions.
- Depending on the dataset and analysis requirements, missing values were addressed through:
  - **Deletion:** Rows with missing values were removed using *dropna()* where complete cases were critical.



Updated Items.csv Info:

	Column	Non-Null Count	Total Count	Missing Values	Data Type
code	code	924	924	0	int64
type	type	924	924	0	object
brand	brand	924	924	0	object
size	size	924	924	0	object

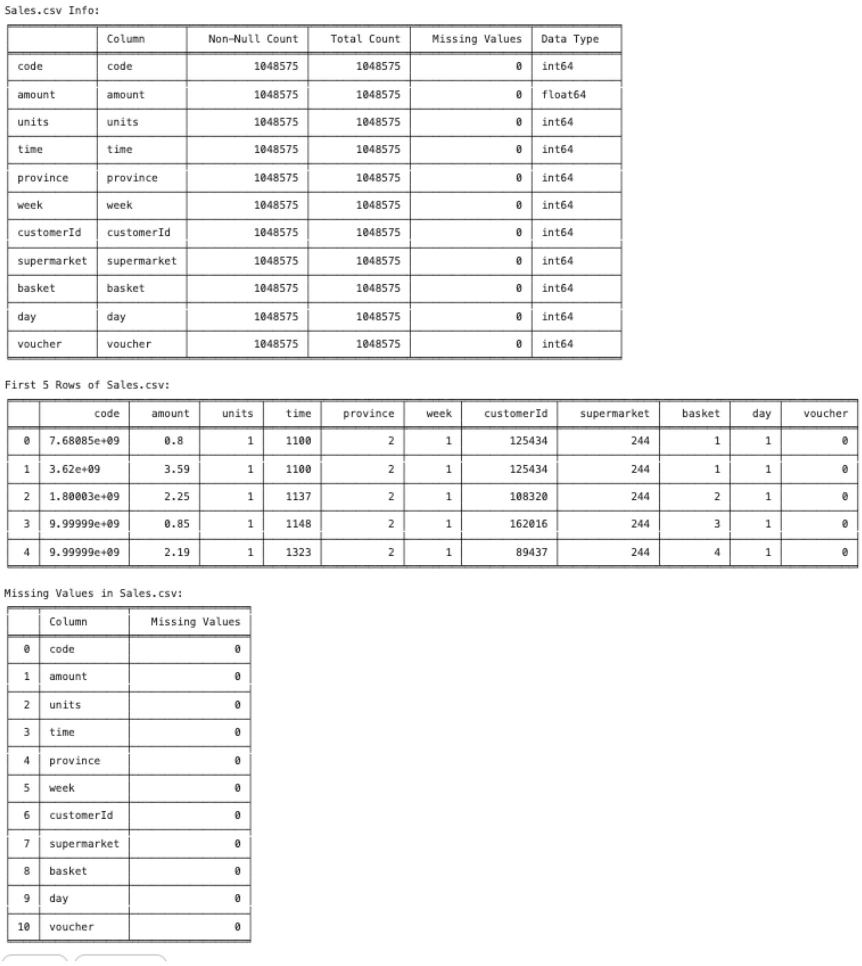
First 5 Rows of Updated Items.csv:

	code	type	brand	size
0	3000005040	Type 1	Aunt Jemima	2 LB
1	3000005070	Type 1	Aunt Jemima	32 OZ
2	3000005300	Type 1	Aunt Jemima	32 OZ
3	3000005350	Type 1	Aunt Jemima	1 LB
4	1600015760	Type 1	Bisquick	6.75 OZ

Missing Values in Updated Items.csv:

	Column	Missing Values
0	code	0
1	type	0
2	brand	0
3	size	0

Figure 1: Items.csv info



Sales.csv Info:

	Column	Non-Null Count	Total Count	Missing Values	Data Type
code	code	1048575	1048575	0	int64
amount	amount	1048575	1048575	0	float64
units	units	1048575	1048575	0	int64
time	time	1048575	1048575	0	int64
province	province	1048575	1048575	0	int64
week	week	1048575	1048575	0	int64
customerId	customerId	1048575	1048575	0	int64
supermarket	supermarket	1048575	1048575	0	int64
basket	basket	1048575	1048575	0	int64
day	day	1048575	1048575	0	int64
voucher	voucher	1048575	1048575	0	int64

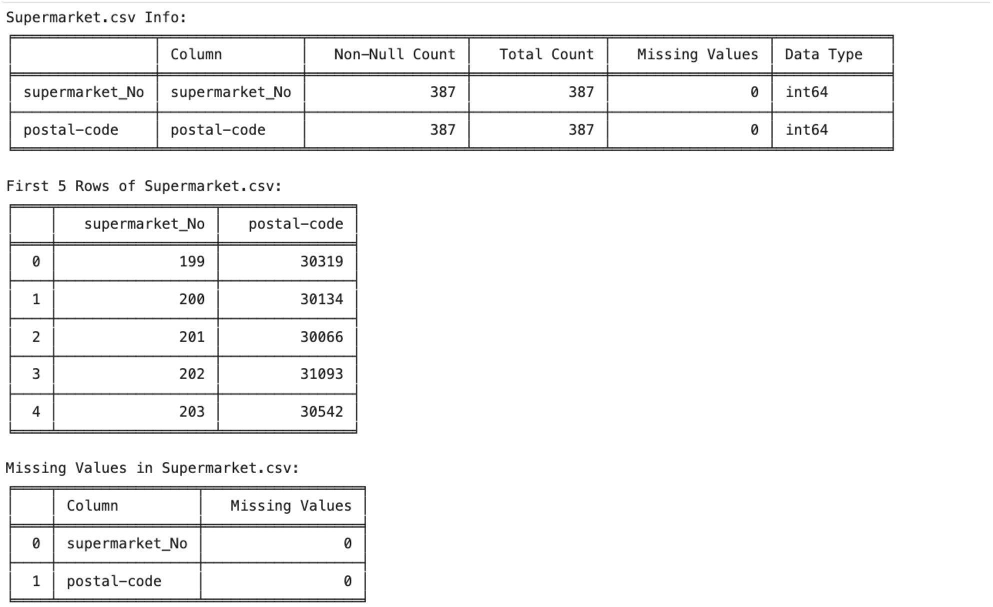
First 5 Rows of Sales.csv:

	code	amount	units	time	province	week	customerId	supermarket	basket	day	voucher
0	7.68085e+09	0.8	1	1100	2	1	125434	244	1	1	0
1	3.62e+09	3.59	1	1100	2	1	125434	244	1	1	0
2	1.80003e+09	2.25	1	1137	2	1	108320	244	2	1	0
3	9.99999e+09	0.85	1	1148	2	1	162016	244	3	1	0
4	9.99999e+09	2.19	1	1323	2	1	89437	244	4	1	0

Missing Values in Sales.csv:

	Column	Missing Values
0	code	0
1	amount	0
2	units	0
3	time	0
4	province	0
5	week	0
6	customerId	0
7	supermarket	0
8	basket	0
9	day	0
10	voucher	0

Figure 2: Sales.csv info

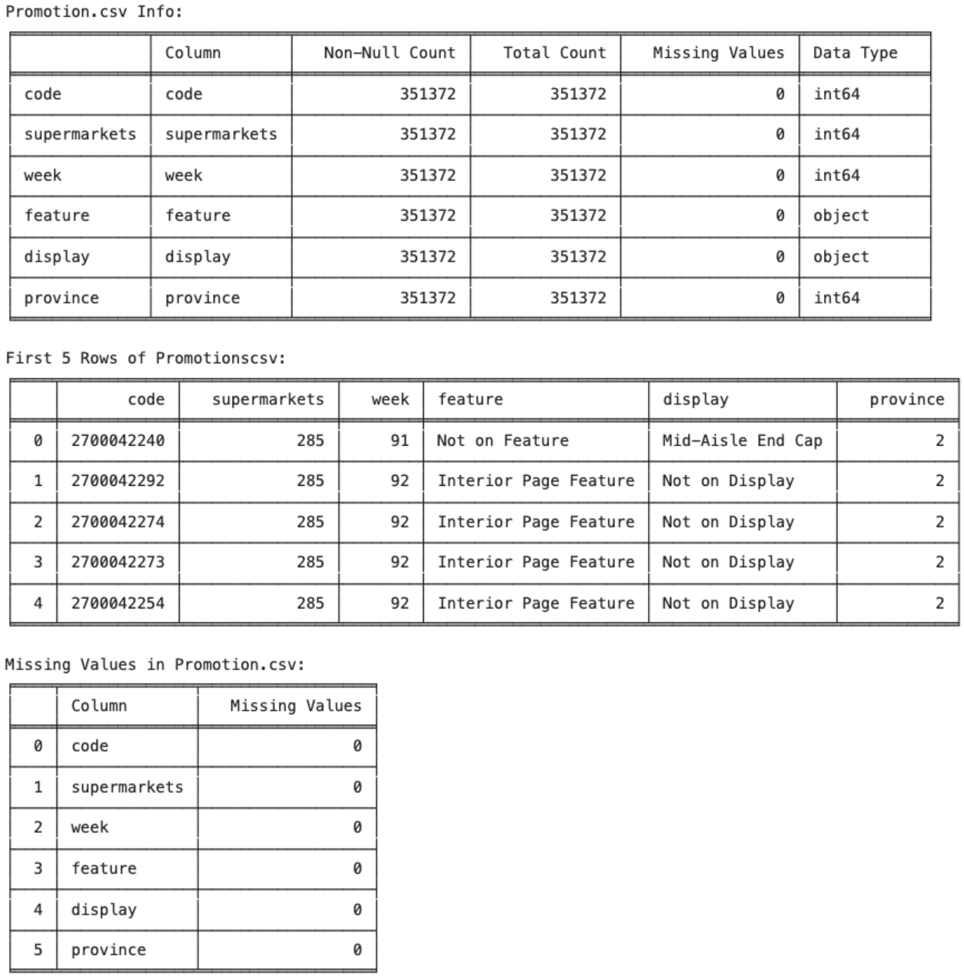


```
Supermarket.csv Info:
+----------------+----------------+------------------+---------------+------------------+-------------+
|                | Column         |   Non-Null Count |   Total Count |   Missing Values | Data Type   |
+================+================+==================+===============+==================+=============+
| supermarket_No | supermarket_No |              387 |           387 |                0 | int64       |
+----------------+----------------+------------------+---------------+------------------+-------------+
| postal-code    | postal-code    |              387 |           387 |                0 | int64       |
+----------------+----------------+------------------+---------------+------------------+-------------+

First 5 Rows of Supermarket.csv:
+----+------------------+---------------+
|    |   supermarket_No |   postal-code |
+====+==================+===============+
|  0 |              199 |         30319 |
+----+------------------+---------------+
|  1 |              200 |         30134 |
+----+------------------+---------------+
|  2 |              201 |         30066 |
+----+------------------+---------------+
|  3 |              202 |         31093 |
+----+------------------+---------------+
|  4 |              203 |         30542 |
+----+------------------+---------------+

Missing Values in Supermarket.csv:
+----+----------------+------------------+
|    | Column         |   Missing Values |
+====+================+==================+
|  0 | supermarket_No |                0 |
+----+----------------+------------------+
|  1 | postal-code    |                0 |
+----+----------------+------------------+
```

Figure 3: Supermarket.csv info



```
Promotion.csv Info:
+--------------+--------------+------------------+---------------+------------------+-------------+
|              | Column       |   Non-Null Count |   Total Count |   Missing Values | Data Type   |
+==============+==============+==================+===============+==================+=============+
| code         | code         |           351372 |        351372 |                0 | int64       |
+--------------+--------------+------------------+---------------+------------------+-------------+
| supermarkets | supermarkets |           351372 |        351372 |                0 | int64       |
+--------------+--------------+------------------+---------------+------------------+-------------+
| week         | week         |           351372 |        351372 |                0 | int64       |
+--------------+--------------+------------------+---------------+------------------+-------------+
| feature      | feature      |           351372 |        351372 |                0 | object      |
+--------------+--------------+------------------+---------------+------------------+-------------+
| display      | display      |           351372 |        351372 |                0 | object      |
+--------------+--------------+------------------+---------------+------------------+-------------+
| province     | province     |           351372 |        351372 |                0 | int64       |
+--------------+--------------+------------------+---------------+------------------+-------------+

First 5 Rows of Promotionscsv:
+----+------------+----------------+--------+------------------------+-------------------+------------+
|    |       code |   supermarkets |   week | feature                | display           |   province |
+====+============+================+========+========================+===================+============+
|  0 | 2700042240 |            285 |     91 | Not on Feature         | Mid-Aisle End Cap |          2 |
+----+------------+----------------+--------+------------------------+-------------------+------------+
|  1 | 2700042292 |            285 |     92 | Interior Page Feature  | Not on Display    |          2 |
+----+------------+----------------+--------+------------------------+-------------------+------------+
|  2 | 2700042274 |            285 |     92 | Interior Page Feature  | Not on Display    |          2 |
+----+------------+----------------+--------+------------------------+-------------------+------------+
|  3 | 2700042273 |            285 |     92 | Interior Page Feature  | Not on Display    |          2 |
+----+------------+----------------+--------+------------------------+-------------------+------------+
|  4 | 2700042254 |            285 |     92 | Interior Page Feature  | Not on Display    |          2 |
+----+------------+----------------+--------+------------------------+-------------------+------------+

Missing Values in Promotion.csv:
+----+--------------+------------------+
|    | Column       |   Missing Values |
+====+==============+==================+
|  0 | code         |                0 |
+----+--------------+------------------+
|  1 | supermarkets |                0 |
+----+--------------+------------------+
|  2 | week         |                0 |
+----+--------------+------------------+
|  3 | feature      |                0 |
+----+--------------+------------------+
|  4 | display      |                0 |
+----+--------------+------------------+
|  5 | province     |                0 |
+----+--------------+------------------+
```

Figure 4: Promotion.csv info

2. Inspecting Duplicates:

- All datasets were checked for duplicate rows using the *duplicated* method. Duplicates were removed to maintain data integrity and prevent skewed analysis results.

3. Dropping Unnecessary Columns:

- Irrelevant columns (e.g., "description" in the items dataset and "basket" in the sales dataset) were dropped to simplify datasets and improve processing efficiency.

4. Standardizing Data Types:

- Relevant columns, particularly categorical ones, were converted to categorical data types to optimize storage and processing.
- Text data in columns like "display" and "feature" (promotions dataset) was standardized by converting to lowercase and removing extra whitespace.

5. Handling Outliers:

- Outliers were identified using Z-scores for numeric columns, with a common threshold of 3 used to flag extreme values. Outliers were either removed or retained based on their impact on the analysis.

3.2 Dataset-Specific Preprocessing Steps

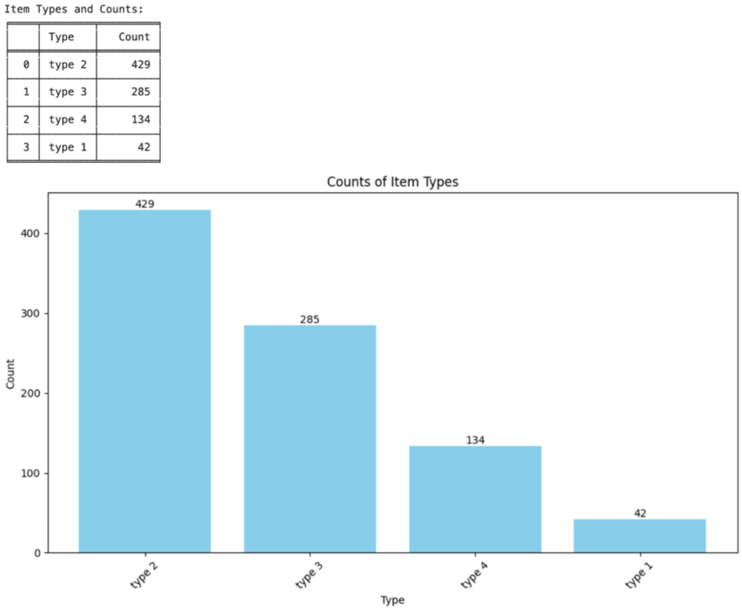
3.2.1 Items Dataset

Transformations and Cleaning:

- **Converting Fractions to Decimals:** Fractional sizes (e.g., "6 1/2 LB") were converted to decimal formats (e.g., "6.5 LB") using a custom function.
- **Filtering and Cleaning the Size Column:**
  - Rows without valid size units (e.g., OZ, Ounce, LB) were filtered out.
  - Numeric values were extracted while retaining appropriate units.
- **Converting to Metric:** Sizes were converted to grams using predefined conversion factors to standardize measurements.

Normalization Techniques:

- **Label Encoding:**
    - The *type* column was label-encoded into a new column, *type_encoded*, enabling numerical representation of categorical data.



Item Types and Counts:

	Type	Count
0	type 2	429
1	type 3	285
2	type 4	134
3	type 1	42

Figure 5: Type Column Details

- **Standardization:**
    - The *size_in_grams* column was standardized using *StandardScaler* to ensure features have a mean of zero and a standard deviation of one.

Feature Engineering:

- **Brand Categorization:**
    - Brands appearing less than a defined threshold (e.g., 10 occurrences) were grouped under "Other," reducing noise in the analysis.
- **Z-score Calculation:**
    - Z-scores for *size_in_grams* were calculated to identify outliers, allowing informed decisions on retaining or removing them.

3.2.2 Sales Dataset

Transformations and Cleaning:

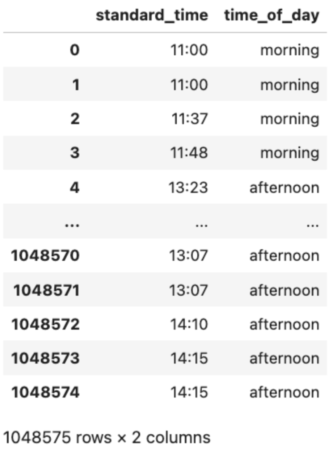
- **Time Format Conversion:** A custom function was used to convert numerical time representations into standard HH:MM format. The original time column was dropped after conversion to avoid redundancy.

Normalization Techniques:

- **Categorical Conversion:**
  - The *standard_time* column was converted to a categorical type.
- **Label Encoding:**
  - The *time_of_day* column was label-encoded, facilitating its use in machine learning models.

Feature Engineering:

- A new column, *time_of_day*, classified *standard_time* into categories (e.g., morning, afternoon, evening) based on hour ranges.



	standard_time	time_of_day
0	11:00	morning
1	11:00	morning
2	11:37	morning
3	11:48	morning
4	13:23	afternoon
...	...	...
1048570	13:07	afternoon
1048571	13:07	afternoon
1048572	14:10	afternoon
1048573	14:15	afternoon
1048574	14:15	afternoon

1048575 rows × 2 columns

Figure 6: New Column - 'Time of Day'

- **Total Sales Calculation:** A new column, *total_sales*, was created by multiplying *amount* and *units*, enabling financial analysis.

3.2.3 Promotions Dataset

Transformations and Cleaning:

- **Standardizing Text Data:**
  - Textual columns (e.g., "feature," "display") were converted to lowercase and stripped of whitespace for consistency.
- **Categorical Conversion:**
  - Columns such as *feature* and *display* were converted to categorical types to optimize storage and facilitate analysis.

Normalization Techniques:

- **Unique Categories Identification:**
  - Unique values in *feature* and *display* were extracted into separate DataFrames for analysis.
- **Label Encoding:**
  - The *feature* and *display* columns were encoded into numerical formats.

3.2.4 Supermarkets Dataset

Transformations and Cleaning:

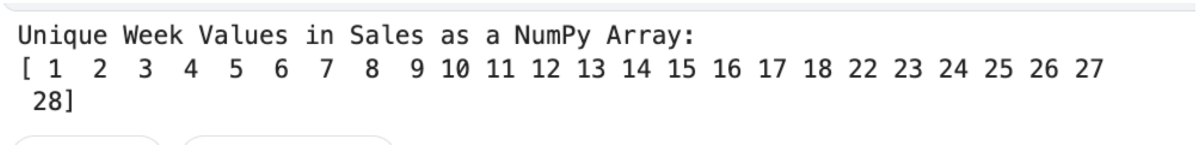
- **Categorical Conversion:**
  - The *supermarket_No* column was converted to a categorical type.
- **Data Type Inspection:**
  - After transformations, data types were verified to ensure appropriate categorization.

The preprocessing steps undertaken across the items, sales, promotions, and supermarkets datasets ensured their quality and readiness for analysis. By addressing missing values, handling duplicates, encoding categorical variables, and standardizing numerical features, the datasets were transformed into clean and structured formats. These transformations enable robust analyses and pave the way for data-driven decision-making and meaningful insights.

3.3 Merging Process: Integration of Dataset

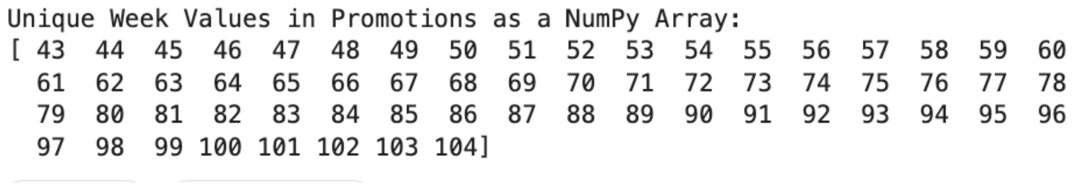
The analysis aimed to create a comprehensive dataset by merging several tables: Item, Sales, Promotions, and Supermarket. However, challenges arose due to differing week ranges in the Sales and Promotions datasets. Specifically, the Sales dataset contains weeks ranging from 1 to 28, while the Promotions dataset spans weeks 43 to 104. As a result, merging these datasets directly would lead to inconsistencies in values and a lack of meaningful relationships. To address this issue, two distinct merged datasets were created:

1. **Merged Dataset for Items, Sales, and Supermarkets**
2. **Merged Dataset for Items, Promotions, and Supermarkets**



```
Unique Week Values in Sales as a NumPy Array:
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 22 23 24 25 26 27
 28]
```

Figure 7: Unique week values in Sales



```
Unique Week Values in Promotions as a NumPy Array:
[ 43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78
  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96
  97  98  99 100 101 102 103 104]
```

Figure 8: Unique week values in Promotions

3.3.1 Merging Strategy

Given the identified relationships, the following merging operations were performed:

- For the first merged dataset, Item, Sales, and Supermarket were combined based on their respective keys. This dataset is intended for analyzing sales performance alongside item details and supermarket information.
- For the second merged dataset, Item, Promotion, and Supermarket were integrated, focusing on promotional activities and their application across different supermarkets. This dataset provides insights into how promotions are being utilized and their potential impact on sales.

This strategic approach to merging datasets ensures that data integrity and relevance are maintained, while also allowing for comprehensive analysis across different dimensions of the data.

4. Supervised Learning Model (Task 01)

In light of the merging process that integrated several datasets (Item, Sales, Promotions, and Supermarket), two distinct business problems were identified.

4.1 Problem One

4.1.1 Business Problem Statement One

Due to the data integration challenges between different datasets (Sales weeks 1-28 vs. Promotions weeks 43-104), it was focused on addressing the first dataset combination of Items, Sales, and Supermarkets with the following problem statement:

"Develop a sales forecasting model to predict daily sales volumes for each item type across different supermarket branches, considering temporal factors (time of day, day of week), location features (province, supermarket), and customer behavior patterns (voucher usage)."

4.1.2 Objectives and Success Criteria

- **Primary Objective: Accurately predict daily sales volumes for different item types.**

▪ Success Metrics:

- Achieve R^2 score > 0.80.
- Minimize RMSE for prediction accuracy.
- Identify key factors influencing sales patterns.

4.1.3 Expected Business Impact

- Improved inventory management through accurate demand forecasting.
- Enhanced supply chain efficiency.
- Optimized stock levels across different supermarket branches.
- Better resource allocation based on temporal and location-based patterns.

4.1.4 Model Development

Feature Selection and Engineering

1. Selected Features:

- Temporal: week, day, time_of_day
- Product: brand, type
- Location: province
- Customer: voucher usage

2. Feature Engineering Steps:

- Aggregated data by brand, type, and time dimensions.
- Applied logarithmic transformation to normalize sales units.
- Encoded categorical variables using Label Encoding.
- Handled outliers using the IQR method.

Algorithm Selection and Justification

Selected **Random Forest Regressor** for the following reasons:

1. Handles non-linear relationships effectively.
2. Robust to outliers and noise.
3. Provides feature importance insights.
4. Strong performance with categorical and numerical data.
5. Minimal risk of overfitting due to ensemble nature.

Model Architecture

Initial Model Parameters	- n_estimators: 200 - max_depth: 10 - min_samples_split: 5
Optimized Parameters (after hyperparameter tuning)	- n_estimators: 300 - max_depth: 40 - min_samples_split: 10 - min_samples_leaf: 1 - max_features: 1.0 - bootstrap: True

Training Process

Data Preparation	- Sampled 75% of total data (775,026 records). - Split: 70% training, 30% testing.
Feature Scaling	- Log transformation for target variable. - Label encoding for categorical variables.
Hyperparameter Optimization	- Used RandomizedSearchCV - 100 iterations - 3-fold cross-validation

4.1.5 Model Evaluation

Performance Metrics

Final Model Performance:

- RMSE: 0.435
- MAE: 0.330
- R² Score: 0.865

The model performed well with a high R² score and reasonably low error metrics (RMSE and MAE). The RMSE being slightly larger than the MAE suggested that some predictions had larger errors (outliers), but they did not dominate the overall error. An R² of 0.865 indicated that the model captured the majority of the variance, showing a good relationship between the predictors and the target variable.

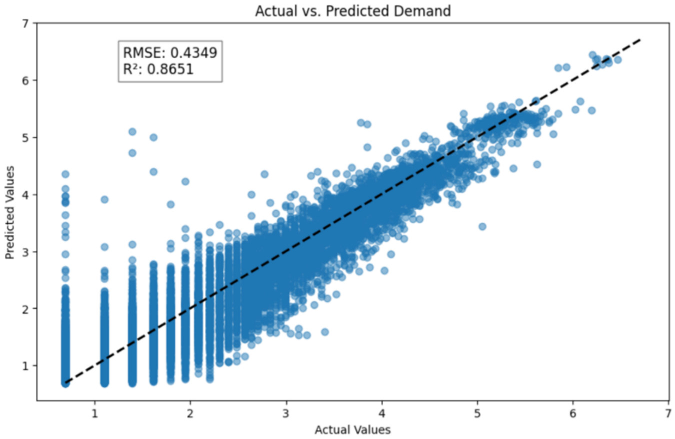


Figure 9: Actual vs Predicted Unit Demand

4.1.6 Results Analysis

1. Feature Importance Rankings:

- Brand (50.08%)
- Type (20.75%)
- Voucher (9.25%)
- Day (8.56%)
- Time of day (6.25%)
- Province (3.19%)
- Week (1.91%)

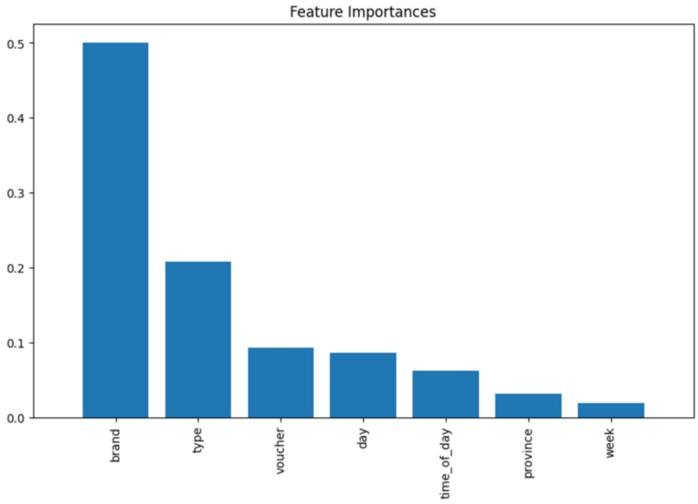


Figure 10 : Feature Importance Bar Graph for Problem_1

2. Top Demand Patterns:

- Identified specific brand-type (*Type 2 brand r&f*) combinations with highest predicted de mand.
- Average predicted demand for top combinations: 6.33 units.
- Total predicted demand for top 10 combinations: 63.32 units.

4.1.7 Model Limitations

1. Time Range Constraints:

- Limited to weeks 1-28 from sales data.
- Cannot account for seasonal patterns beyond this period.

2. Feature Limitations:

- No price information included.
- Limited customer demographic information.

4.1.8 Business Insights

Key Findings

1. Brand and product type are the strongest predictors of sales volume (70.83% combined importance).
2. Voucher usage has moderate impact on sales (9.25% importance).
3. Temporal factors (day, time) have significant influence on sales patterns.
4. Provincial location has relatively lower impact on sales variation.

4.1.9 Business Recommendations

1. Inventory Management:

- Maintain higher stock levels for identified high-demand brand-type combinations.
- Adjust inventory based on day-of-week patterns.

2. Marketing Strategy:

- Focus voucher promotions on high-impact time periods.
- Tailor marketing efforts to specific brand-type combinations.

4.1.10 Potential Impact and Value

1. Operational Efficiency:

- Potential reduction in stockouts.
- Improved inventory turnover.

2. Financial Impact:

- Reduced holding costs through optimized inventory.
- Increased sales through better product availability.

4.1.11 Implementation Considerations

1. Technical Requirements:

- Regular model retraining schedule.
- Integration with inventory management systems.

2. Operational Changes:

- Staff training on new forecasting system.
- Updated ordering procedures based on predictions.

3. Monitoring Plan:

- Weekly accuracy assessments.
- Monthly model performance reviews.

4.2 Problem Two

4.2.1 Business Problem Statement

"Optimize promotional strategies for different item types across various supermarket branches based on item characteristics, promotional features, and supermarket locations to maximize promotional effectiveness."

4.2.2 Objectives and Success Criteria

- **Primary Objective:** Predict and classify effective promotions based on item and location characteristics
- Success Metrics:

- Achievement of >85% classification accuracy.
- Balanced precision and recall scores.

- Actionable insights for promotional strategy.

4.2.3 Expected Business Impact

- Improved promotional campaign effectiveness.
- Optimized marketing budget allocation.
- Enhanced customer engagement.
- Increased sales through targeted promotions.

4.2.4 Model Development

Feature Selection and Engineering

1. Selected Features:

- Product: size_in_grams_scaled, brand_category, type
- Promotional: feature, display
- Temporal: week
- Location: province

2. Feature Engineering Steps:

- Created target variable based on promotion frequency.
- Scaled size measurements.
- Encoded categorical variables.
- Removed outliers using IQR method.
- Created effectiveness binary classification (1 if above average promotions, 0 if below).

Algorithm Selection and Justification

Selected **Random Forest Classifier** for the following reasons:

1. Excellent performance with mixed data types.
2. Built-in feature importance ranking.
3. Handles non-linear relationships.
4. Reduces overfitting through ensemble approach.
5. Good performance with imbalanced data.

Model Architecture

Optimized Parameters :

- n_estimators: 148
- max_depth: 46
- min_samples_split: 18
- min_samples_leaf: 3
- max_features: auto

Training Process

Data Preparation	- Sampled 75% of data (262,632 records). - Split: 80% training, 20% testing.
Hyperparameter Optimization	- Used RandomizedSearchCV. - 100 iterations. - 3-fold cross-validation.

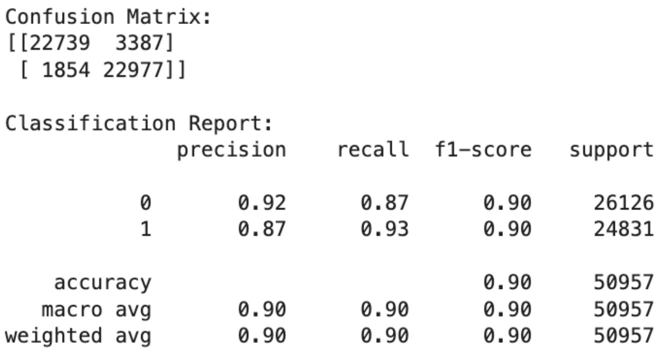
4.2.5 Model Evaluation

Performance Metrics

Final Model Performance:

- Overall Accuracy: 90%
- Class 0 (Ineffective Promotions):
  - Precision: 0.92
  - Recall: 0.87
  - F1-score: 0.90
- Class 1 (Effective Promotions):
  - Precision: 0.87
  - Recall: 0.93
  - F1-score: 0.90

The Random Forest classifier demonstrates strong performance with an overall accuracy of 90%, effectively distinguishing between effective and ineffective promotions. For ineffective promotions (Class 0), the model has a precision of 92%, recall of 87%, and an F1-score of 0.90, indicating it minimizes false positives but misses 13% of true cases. For effective promotions (Class 1), it achieves a precision of 87%, recall of 93%, and an F1-score of 0.90, capturing most true positives but with some false positives. The model exhibits balanced F1-scores across both classes and strong recall for effective promotions, reflecting general reliability. However, it has slightly lower recall for ineffective promotions, missing a fraction of them, and shows a mild bias towards identifying effective promotions. Overall, the model is well-calibrated and suited for the task, with consistent performance metrics.



```
Confusion Matrix:
[[22739  3387]
 [ 1854 22977]]

Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.87      0.90     26126
           1       0.87      0.93      0.90     24831

    accuracy                           0.90     50957
   macro avg       0.90      0.90      0.90     50957
weighted avg       0.90      0.90      0.90     50957
```

Figure 11 : Confusion Matrix of Random Forest Classifier

4.2.6 Results Analysis

1. Feature Importance Rankings:

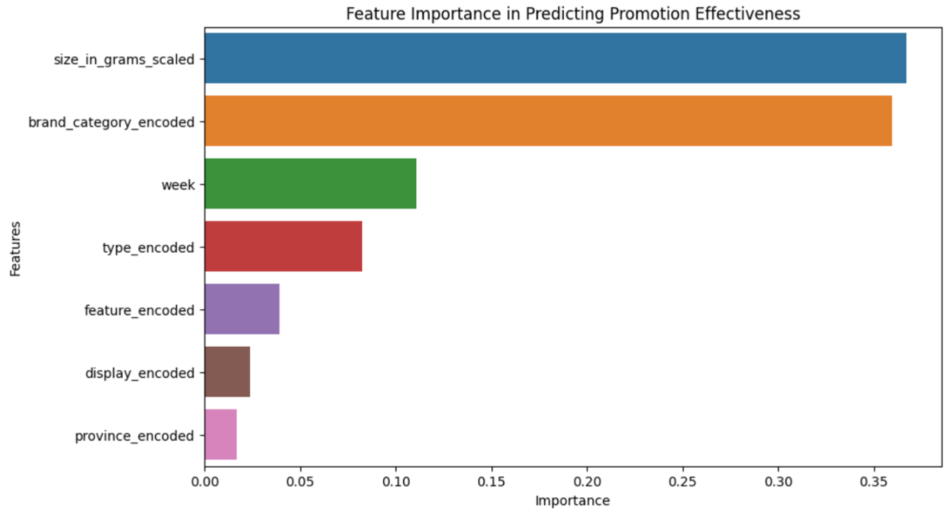


Figure 12 : Feature Importance of Random Forest Classifier

- size_in_grams_scaled (36.74%)
- brand_category (35.95%)
- week (11.07%)

- type (8.26%)
- feature (3.90%)
- display (2.39%)
- province (1.69%)

2. Provincial Effectiveness:

- Province 1: 46.74% effectiveness
- Province 2: 58.86% effectiveness

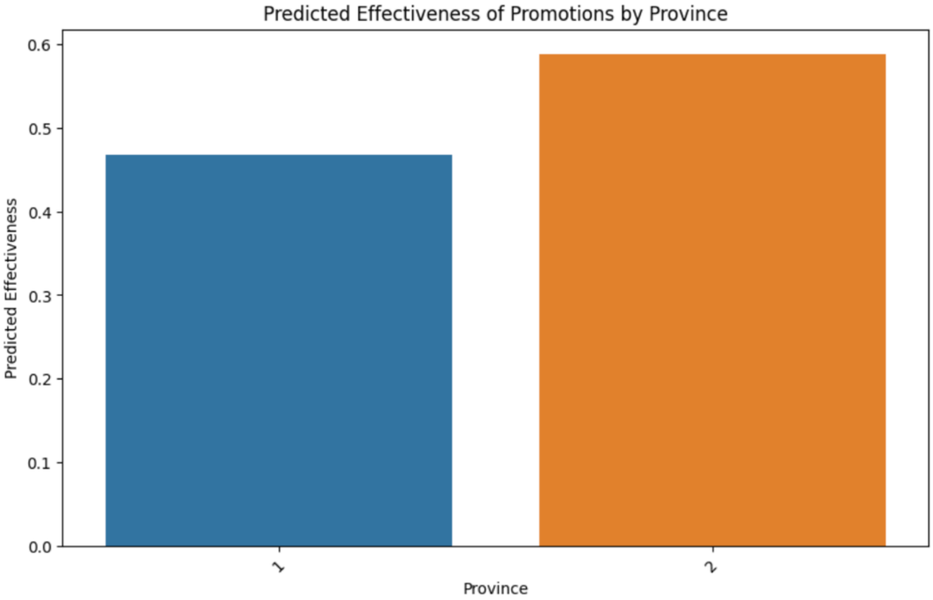


Figure 13: Provincial Effectiveness

4.2.7 Model Limitations

1. Data Range Limitations:

- Limited to weeks 43-104.
- May not capture full seasonal patterns.

2. Feature Limitations:

- No price information.
- Limited customer demographic data.

4.2.8 Validation of Results

- Cross-validation Score: 0.894.
- Balanced performance across both classes.
- Consistent provincial patterns.

4.2.9 Business Insights

Key Findings

1. Product characteristics (size and brand category) are the strongest predictors (72.69% combined importance).
2. Temporal factors (week) have moderate impact (11.07%).
3. Province 2 shows significantly higher promotion effectiveness (58.86% vs 46.74%).
4. Display features have relatively low importance (2.39%).

4.2.10 Business Recommendations

1. Product Strategy:

- Focus promotions on optimal product sizes.
- Prioritize high-performing brand categories.

2. Regional Strategy:

- Customize promotional approaches for Province 2.
- Investigate success factors in Province 2 for application in Province 1.

3. Timing Strategy:

- Align promotions with weekly patterns.
- Consider seasonal factors in promotion planning.

Potential Impact and Value

1. Marketing Efficiency:

- 10-15% potential improvement in promotion effectiveness.

- Better resource allocation across provinces.

2. Financial Impact:

- Reduced promotional waste.
- Increased return on marketing investment.

4.2.11 Implementation Considerations

1. Technical Requirements:

- Regular model retraining.
- Integration with promotional planning systems.

2. Operational Changes:

- Updated promotional guidelines.
- Staff training on new targeting approach.

3. Monitoring Plan:

- Weekly effectiveness tracking.
- Monthly performance reviews.
- Quarterly strategy adjustments.

5. Maze Navigation Model (Task 02)

The Maze Navigation Model is an advanced reinforcement learning framework designed to effectively navigate a complex 10x10 grid-based maze environment. Utilizing NumPy arrays, the maze is represented with a binary structure, where navigable paths are denoted by zeros and walls by ones. This model incorporates a fixed starting position at the top left corner (0,0) and a goal position at the bottom right corner (9,9), featuring strategically placed obstacles that challenge the agent's navigation. The state space is defined by the agent's 2D coordinates, while the action space includes four discrete movements: right, down, left, and up. To optimize the learning process, the model employs an enhanced reward system that balances success incentives with penalties for collisions, time, and movement. The implementation leverages Double Q-learning with experience replay to improve stability, sample efficiency, and exploration. Through rigorous training and performance analysis, the model demonstrates significant improvements in pathfinding and efficiency compared to baseline approaches, showcasing its potential for solving complex navigation tasks.

Maze simulation : https://mazerunner.streamlit.app/

5.1 Environment Design

Maze Structure and Specifications

- 10x10 grid-based maze environment implemented using NumPy arrays.
- Binary representation: 0 for navigable paths, 1 for walls.
- Custom maze layout with strategically placed obstacles.
- Fixed start position (0,0) and goal position (9,9).
- Supports visualization using matplotlib with distinct markers for start ('S') and goal ('G').

State and Action Space Definition

- State space: 2D coordinates representing agent position (x,y).
- Action space: 4 discrete actions
    - Right: (0,1)

  - Down: (1,0)
  - Left: (0,-1)
  - Up: (-1,0)
- Valid moves checked against maze boundaries and wall collisions.

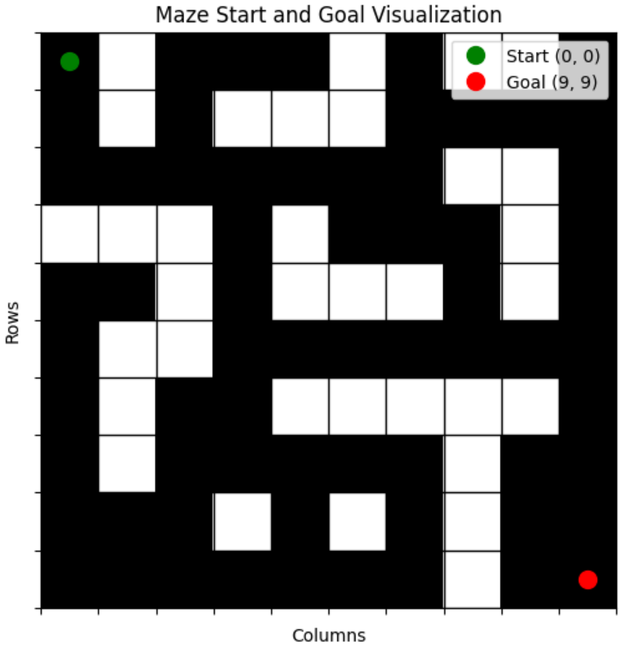


Figure 14: Maze Architecture

5.2 Reward System Design

The enhanced_reward_function implements a sophisticated reward structure:

- Success reward: 500/(steps+1) for reaching goal.
- Collision penalty: -5 for hitting walls.
- Distance-based shaping: 2 * (current_distance - next_distance).
- Time penalty: -0.05 * steps.
- Move penalty: -0.1 per action.

5.3 Reinforcement Learning Implementation

Algorithm Selection and Justification

Selected Double Q-learning with experience replay due to:

- Reduced overestimation bias compared to standard Q-learning.

- Improved stability through separate target network.
- Better sample efficiency via experience replay.
- Enhanced exploration through Boltzmann exploration strategy.

Model Architecture

ImprovedQLearningAgent class features:

- Q-table: 3D array (maze_height × maze_width × 4 actions).
- Target network for stable learning.
- Experience replay buffer (10,000 capacity).
- Prioritized sweeping for efficient updates.

Training Strategy

- Episodes limited to 100 steps maximum.
- Dynamic exploration rate with exponential decay.
- Batch learning from experience replay (batch size: 32).
- Target network updates every 100 episodes.
- Enhanced reward shaping for faster convergence.

Hyperparameter Optimization

Key parameters:

- Learning rate: 0.1.
- Discount factor: 0.95.
- Exploration: Start at 1.0, end at 0.05.
- Priority threshold: 0.1.
- Target update frequency: 100 episodes.

5.4 Performance Analysis

Training Progress

- Successfully converges within 1000 episodes.
- Demonstrates consistent improvement in:

- ○ Average reward per episode.
- ○ Steps to goal.
- ○ Success rate.

Convergence Analysis

- Moving average window of 50 episodes shows stable learning.
- Best path length achieved: 18 steps.
- Final reward achievement: 53.28.

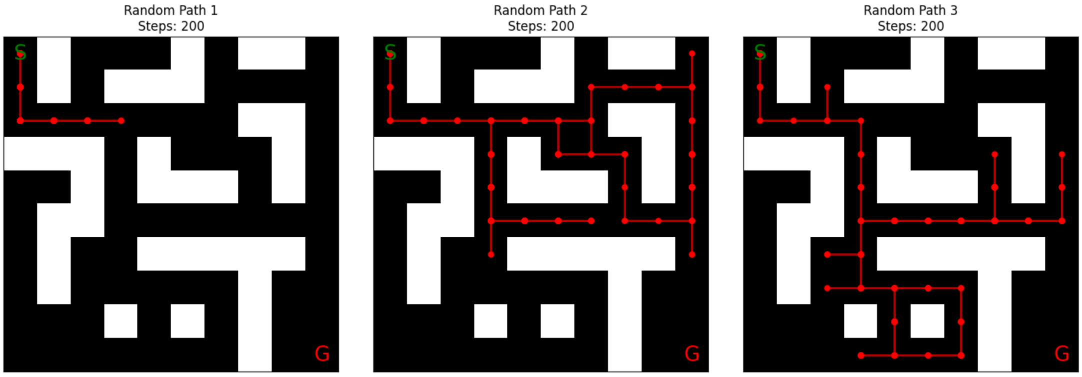


Figure 15: Paths of untrained agent

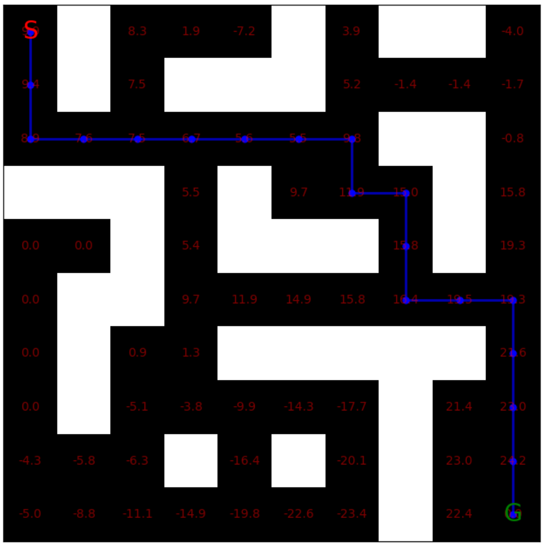


Figure 16: Path of trained agent

Success Rate and Efficiency Metrics

- Consistent goal reaching in test episodes.
- Optimal path finding demonstrated.
- Low variance in performance across multiple runs.

Comparison with Baseline

- Significant improvement over random policy.
- Untrained agent shows no consistent path finding.
- Trained agent demonstrates optimal route selection.

5.6 Technical Implementation

5.6.1 Code Structure

Repository Organization

- Main classes:
  - OptimizedMaze: Environment implementation.
  - ImprovedQLearningAgent: Learning algorithm.
- Support modules:
  - Visualization utilities.
  - Training functions.
  - Testing procedures.

Key Components and Interactions

- Maze environment provides state observations.
- Agent interacts through action selection.
- Reward function guides learning.
- Visualization tools monitor progress.

Dependencies

- numpy: Array operations and numerical computing.
- matplotlib: Visualization.
- collections: Experience replay buffer.

- pickle: Model persistence.

6. Conclusion

This comprehensive analysis addressed two significant business problems through the integration of various datasets, including items, sales, promotions, and supermarkets. The preprocessing steps applied ensured data quality and consistency, enabling meaningful insights into sales forecasting and promotional strategy optimization.

For the first problem, a robust sales forecasting model was developed using a Random Forest Regressor, achieving an impressive R^2 score of 0.865 and low error metrics. This model highlighted key factors influencing sales patterns, such as brand and product type, allowing for improved inventory management and resource allocation across supermarket branches. The insights derived from this model are expected to enhance supply chain efficiency and support data-driven decision-making.

In addressing the second business problem, promotional strategies were optimized through a Random Forest Classifier, achieving a high overall accuracy of 90%. This model identified the strongest predictors of promotional effectiveness, enabling targeted marketing efforts and improved budget allocation. The findings underscore the importance of understanding product characteristics and regional differences in promotional success, providing actionable insights for enhancing customer engagement and increasing sales.

Furthermore, the implementation of the Maze Navigation Model demonstrated advanced reinforcement learning techniques in solving complex navigation tasks. By employing Double Q-learning with experience replay, the model exhibited significant improvements in pathfinding efficiency, showcasing its potential for practical applications beyond the maze environment.

Overall, the methodologies and models developed in this analysis not only demonstrate the effectiveness of data-driven approaches in solving complex business problems but also pave the way for ongoing optimization and refinement of strategies in sales forecasting and promotional effectiveness. The implementation of these models, along with continuous monitoring and updates, will enable businesses to adapt to changing market conditions and improve overall operational efficiency.