## **Task Overview**

**TASK 1**

**The project involves**:

1. **Data Cleaning & Transformation** – Standardizing, handling missing values, and merging datasets.
2. **Feature Engineering** – Creating useful features for analysis.
3. **Supervised Learning for Sales Forecasting** – Predicting sales performance using RandomForestRegressor.
4. **Supermarket Performance Classification** – Identifying high-performing supermarkets using RandomForestClassifier.

## **Data Cleaning & Transformation**

**Steps Taken:**

* Removed duplicates.
* Standardized column names for consistency.
* Trimmed whitespace from string values.
* Filled missing values with meaningful defaults.
* Converted relevant columns to appropriate data types (e.g., dates, floats, and integers).
* Merged datasets on key columns (e.g., code, Supermarket\_No, province).

**Challenges Faced:**

* Missing or inconsistent data formats.
* Need for additional feature engineering to capture sales patterns.

## **Supervised Learning Model: Sales Forecasting**

**Problem Definition:**

* Predict **sales amount** based on promotions, store location, and product features.

**Model Choice:**

* **RandomForestRegressor** was selected due to its robustness in handling non-linear relationships and categorical data.

**Features Used:**

* **Product-related:** code, brand, type
* **Store-related:** Supermarket\_No
* **Time-related:** Month, Weekday
* **Promotion-related:** Promotion Active

**Evaluation Metrics:**

* Root Mean Squared Error (RMSE) was used to measure model performance.
* Achieved an RMSE score indicating model accuracy.

**Insights & Business Value:**

* Promotion effectiveness analysis identified key periods where sales were positively impacted.
* High-performing products and stores were identified, aiding inventory planning.

## **Supervised Learning Model: Supermarket Performance Classification**

**Problem Definition:**

* Classify supermarkets into **high-performing (1)** and **low-performing (0)** based on total sales.

**Model Choice:**

* **RandomForestClassifier** was used due to its ability to handle categorical variables and rank feature importance.

**Features Used:**

* Month, Weekday, code, Promotion Active, Display Impact

**Evaluation Metrics:**

* Accuracy was used to measure classification performance.
* Achieved high accuracy, confirming that key features impact supermarket performance.

**Insights & Business Value:**

* Helped in identifying **top-performing supermarkets**, guiding promotional strategies.
* Understanding of key sales drivers enabled targeted marketing efforts.

## **Business Insights**

**Key Findings:**

* **Promotion Impact:** Sales were significantly higher when promotions were active.
* **Weekend Sales Surge:** More transactions occurred on weekends.
* **High-Performing Stores:** Some supermarkets consistently outperformed others due to location and promotions.
* **Customer Segmentation:** High spenders contributed disproportionately to sales.

**Methodologies Used:**

* Data aggregation and visualization.
* Feature importance analysis.
* Predictive modeling to support decision-making.

**Business Relevance:**

* Improved **inventory management** by predicting demand.
* Optimized **marketing strategies** based on customer spending behavior.
* Enhanced **store operations** through data-driven decisions.

**---------------------------------------------------------------------------------------------------------------------**

**TASK 2**

**Maze Model Design**

**Explanation of the Maze Navigation Model**

The maze is represented as a 10x10 NumPy array, where different values indicate different elements:

* **-1**: The starting position of the agent.
* **0**: Open paths where movement is allowed.
* **1**: Walls that act as obstacles.
* **2**: The goal position.

The agent navigates through the maze using four possible actions: **up, down, left, and right**. The movement is restricted by walls and boundaries. If an agent attempts to move into a wall, it remains in the same position and receives a negative reward.

**Overview of the Reinforcement Learning Approach**

The model employs **Q-learning**, a reinforcement learning algorithm based on the Bellman equation. Key components include:

* **Q-table**: A matrix storing Q-values for each state-action pair.
* **Exploration vs. Exploitation**: The agent selects actions using an epsilon-greedy policy, balancing exploration (random actions) and exploitation (choosing the best-known action).
* **State Transition**: Based on the selected action, the agent moves to a new state if the movement is valid.
* **Rewards**:
  + Reaching the goal yields **+100**.
  + Moving to a valid path results in **-1**.
  + Attempting to move into a wall incurs a penalty of **-10**.
* **Q-value Update**:
  + The Q-value for the selected action is updated using: where is the learning rate and is the discount factor.

**Training Results and Performance Analysis**

* The training was run for **30 iterations**.
* Each iteration starts from the initial position and attempts to reach the goal using learned Q-values.
* The number of steps required to reach the goal was tracked over iterations.
* The results indicate a **decreasing trend in steps**, signifying successful learning.
* The learning curve, plotted with **iterations vs. steps to goal**, shows that over time, the agent improves its navigation efficiency.
* Visualization of the maze during training helps observe the agent's learning progress in real-time.