Dynamic Pricing for Food Waste Reduction in Supermarkets Using Reinforcement Learning

MSc Research Project Part 2
MSc. in Data Analytics

Sathwik Kataradahalli Sumanth
Student ID: x23416823

School of Computing
National College of Ireland

Supervisor: Prof. Eamon Nolan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Sathwik Kataradahalli Sumanth		
Student ID:	x23416823		
Programme:	MSc. in Data Analytics	**Year:**	2025 - 2026
Module:	MSc Research Project Part 2		
Supervisor:	Prof. Eamon Nolan		
Submission Due Date:	28/01/2026		
Project Title:	Dynamic Pricing for Food Waste Reduction in Supermarkets Using Reinforcement Learning		
Word Count:	7715	**Page Count:**	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Sathwik Kataradahalli Sumanth

Date: 28/01/2026

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Dynamic Pricing for Food Waste Reduction in Supermarkets Using Reinforcement Learning

Sathwik Kataradahalli Sumanth
x23416823

Abstract
Supermarkets usually use a fixed price or a plain markdown scheme to sell perishable goods, and thus they lose either too much or too little money. Reinforcement learning (RL) is a data-driven alternative that can adjust the prices according to the demand and inventory changes. This paper creates a simple and reproducible simulator that simulates FEFO stock ageing, stochastic demand and discrete price Multipliers, and determines whether a policy-gradient RL approach, Proximal Policy Optimization (PPO), can learn to perform better than heuristic knowledge of the supermarket. Three policies (Fixed, Random and PPO) are evaluated on a series of simulated 589-day episodes with a reward function which specifically punishes expired stock. On five randomly chosen seeds, PPO will always get to zero waste and competitive revenue with the greatest total payoff despite the fact that it has a conservative operation as compared to the Fixed and Random policies depending on the waste penalty. The Random policy has erratic performance and average waste whereas the Fixed baseline maximizes revenue but results in extreme waste which is operationally impossible. These results indicate that even a naive RL policy can encode perishability constraints and be more helpful in providing a better revenue-waste tradeoff than the usual retail heuristics. The future work must ensure that it has more demand modelling, continuous pricing and operations constraints to ensure that it can be deployed in real supermarket environments.
Keywords - Reinforcement learning, Proximal Policy Optimization (PPO), dynamic pricing, perishable goods, retail supermarket, FEFO inventory ageing, waste reduction, sustainability, revenue management, stochastic demand, Gymnasium simulator, stable-baselines3

1 Introduction

1.1 Background

One of the major environmental, economic, and operational issues of supermarkets is food waste. Perishable goods tend to go into expiry before they are sold due to a lack of dynamism in pricing and clearance policies to react to demand states and shelf life left. The retailers generally use either fixed price or simple markdown policies like using a fixed percentage discount when expiry is near, but these policies do not consider age of inventory, demand variation or real-time selling situation. This often compels the supermarkets to decide between deep end of life discounts that undermine the profit margins and non-sold stock that is a direct contributor to food wastage.

The possible advantages of more adaptive control strategies have been proven by a considerable amount of research that has been performed on perishable inventory management and dynamic pricing. Based on analytical and simulation-based research, the interplay between perishability, policies of issuing, shelf-life stages, and price-sensitive demand is studied and seeks to balance levels of revenue, service, and reduction of spoilage (Ding and Peng, 2024; Fan et al., 2020; Moshtagh et al., 2025; Rios and Vera, 2023; Solari et al., 2024). Other literature comprises competitive pricing situations (Kastius and Schlosser, 2021) and literature where customer learning impacts demand (Vahdani and Sazvar, 2022), displaying the relationship in which pricing decisions can spread via customer expectations.

Recent developments in reinforcement learning (RL) and deep RL (DRL) provide data-driven approaches to the derivation of pricing and ordering policies, grounded on interaction with an environment, instead of only using closed-form optimization models. DRL has been used in joint pricing-inventory control to perishables, vendor-managed inventory setting, and strategic-customer setting (Kavoosi et al., 2025; Mohamadi et al., 2024; Nomura et al., 2025; Alamdar and Seifi, 2024; Yavuz and Kaya, 2024; Zhou et al., 2022). These analyses all conclude that dynamic pricing is capable of enhancing business operation performance in situations where perishability, price elasticity, and inventory ageing are clearly modeled.

Nevertheless, a large part of the literature available is given to profit or cost optimization. Reduction of food waste, although of practical necessity, is rather regarded as a by-product, and not as a design goal. This introduces a disparity between the sophisticated RL pricing models and the demands of the supermarket managers who must operate within the high sustainability pressures and demand policies that minimize waste but still generate revenue.

1.2 Importance and Motivation

Supermarkets are under pressure to reduce food waste and still make profits. Poor pricing choices, late markdowns, and non-responsive clearance policies are a contributor to out-of-date inventory, lost sales, and increased disposal expenses. An alternative to this is offered by RL: it is able to learn pricing policies that react directly to indicators of perishability, including inventory age, stochastic demand, and real-time sales performance.

A previous work of DRL illustrates that it leads to increased multi-period prices, ordering policy, and perishability-based decision-making (Kavoosi et al., 2025; Mohamadi et al., 2024; Nomura et al., 2025; Alamdar and Seifi, 2024; Yavuz and Kaya, 2024; Zhou et al., 2022). Nevertheless, the real-life applications in supermarkets demand models that are understandable, reproducible, and can take advantage of comparatively straightforward state variables, including the underlying demand and the remaining shelf life, without using very complex or high-frequency customer behavioral information.

This research will thus consider the question of whether a reinforcement learning strategy can be used to optimize the revenue-waste trade off of a perishable product family in a retail simulation. The RL agent is pitted against two intuitive base strategies, which are fixed pricing and random pricing, to determine the existence of quantifiable operational advantages of learning-based decision-making.

The research question to be answered in the study is: **What is the optimal way to create a reinforcement learning-based dynamic pricing framework to minimize food waste and maximize revenue in supermarkets that sell perishable goods?**

The study aims to answer this question by taking four objectives:

1. Examine the state of the art in perishable inventory management and dynamic pricing, focusing on the techniques of RL/DRL and their limitations (Ding and Peng, 2024; Fan et al., 2020; Kastius and Schlosser, 2021; Kavoosi et al., 2025; Mohamadi et al., 2024; Moshtagh et al., 2025; Nomura et al., 2025; Alamdar and Seifi, 2024; Rios and Vera, 2023; Solari et al., 2024; Vahdani and Sazvar, 2022; Yavuz and Kaya, 2024; Zhou et al., 2022).
2. Develop a simulation environment, which models the inventory ageing, expiry, stochastic demand as well as the daily pricing decisions.
3. Train a reinforcement learning policy (PPO) and compare the performance with the fixed and random pricing baselines.
4. Measure the stability and temporal behavior of episode-level revenue, waste and reward measures by assessing all policies using multiple seeds.

The value of the work assumes the evidence of a waste-sensitive RL pricing model in that perishability can be explicitly modeled, and determining whether RL can be successful in achieving high-level operational performance over intuitively driven supermarket heuristics.

1.3 Limitations and Assumptions

This research is done in a simulated supermarket as opposed to a real-life retail store. Factors that are real in the world, like the concept of cross-product substitution, competitor response, high-resolution footfall information, or customer learning impacts (Kastius and Schlosser, 2021; Alamdar and Seifi, 2024; Vahdani and Sazvar, 2022) are not captured. The simulator concentrates on one of the store-products families and the demand is modeled using base demand and level of promotion as opposed to more elaborate behavioral models. The employed RL algorithm is Proximal Policy Optimization (PPO) as a policy-gradient algorithm, as opposed to the Q-learning or DQN methods a few studies discussed (Kavoosi et al., 2025; Nomura et al., 2025; Yavuz and Kaya, 2024; Zhou et al., 2022). The training and evaluation are fully offline with the aid of several simulated seeds and additional efforts would be necessary to test the policy on a real operation.

Key assumptions include:
(i) perishability could be modelled through FEFO inventory ageing and expiry buckets;
(ii) demand decreases monotonically with price,
(iii) a penalty-based reward function is a suitable proxy of waste expenses and
(iv) a multi-year period of simulation is a reasonable approximation of retail behavior.

The assumptions are aligned with previous research in perishable inventory modelling and RL-based pricing (Ding and Peng, 2024; Fan et al., 2020; Kastius and Schlosser, 2021; Kavoosi et al., 2025; Mohamadi et al., 2024; Moshtagh et al., 2025; Nomura et al., 2025; Alamdar and Seifi, 2024; Rios and Vera, 2023; Solari et al., 2024; Vahdani and Sazvar, 2022; Yavuz and Kaya, 2024; Zhou et al., 2022).

1.4 Structure of the Report

The rest of this report will be divided into six chapters. Chapter 2 is a review of classical optimization and behavioral models to perishable pricing, then reinforcement learning and deep reinforcement learning methods and identifies the gap in the research that this study fills. Chapter 3 defines the research methodology, such as the general structure, data preparation process, creation of the simulation environment, the description of the baseline and RL policies, and the measurement framework to test performance in a variety of seeds. Chapter 4 gives the design specification of the system architecture, which includes data preparation layer, FEFO-based gymnasium environment and the reinforcement learning and the base elements. Chapter 5 explains how the data preprocessing, environment modules, training of the PPO model, and multi-seed evaluation are done along with the tools and libraries used. The experimental findings in Chapter 6 discuss the results of three case studies such as demand and promotion analysis, single-episode policy comparison, and multi-seed robustness evaluation, which is backed by quantitative metrics, visual comparisons and distributional analysis. Chapter 7 finally provides a summary of the main findings, the implications of the findings on retail pricing, and recommendations on how the research and operational deployment should be conducted in the future.

2 Related Work

Pricing and inventory management studies of perishable goods may be generally divided into two complementary streams: (1) classical and behavioral models, which are structured optimization problems which make stylized assumptions. (2) reinforcement learning (RL) and deep RL (DRL) models, which are learning pricing choices without using closed-form demand models through simulated interaction only.

2.1 Classical & Behavioral Models for Perishable Pricing and Inventory

Classical optimization models examine pricing and replenishment in the context of perishability based on clear assumptions of demand elasticity, issuance policies and cost designs. Ding and Peng (2024) prove that the issuance policy (FIFO or FEFO or both) has a significant impact on the age of inventory and, consequently, the spoilage risk. They highlight that it is important to explicitly model the age of inventory- a concept applied in the FEFO-based environment in this project.

In the study conducted by Fan, Xu and Tao (2020), the authors come up with a joint pricing-replenishment of fresh produce when the pricing choices are in contact with the demand elasticity, lead times and left shelf life. In their work, they emphasize the underlying sell-through versus margin trade-off of the retailer. Despite the fact that these structural models are interpretable and provide valuable managers with information, they are weak when promotions or seasonality are involved due to their stylized assumptions.

More comprehensive models of operation are those that look at cross-store and cross-product interactions. Rios and Vera demonstrate that the effect of cross-item substitution, cannibalization and stock transfers drastically increases the dimensionality of retail chains, making the traditional optimization hard to calibrate (2023). These observations encourage complex environment simulation-based or learning-based controllers.

Markdown strategies are also illuminated through the perishable inventory research that is simulation based. Solari et al. (2024) analyzes the demand and discount policy under price sensitivity and on a discrete time simulation determines how early and late markdowns influence the waste-margin frontier. These models assist in managerial what-if analysis but are not flexible to the dynamic demand changes.

A number of works represent perishability as age stages. Moshtagh, Zhou and Verma (2025) differentiate fresh, ageing and near-expiry phases and reveal that price sensitivity is stronger as the left shelf life reduces. Although operationally viable, multi-stage modelling increases the state space by far, and is difficult to estimate due to limited data.

Classical approaches are also supplemented by behavioral models. Vahdani and Sazvar (2022) integrate social-learning effects in price decisions of online perishables, which gives an example of how customer behavior and influence of the peer change the demand formation. These dynamics of behavior are outside the area of this project but reflect the real-life complexities that are not accommodated in the traditional models.

Summary:

Classical and behavioural models give a structural insight into the dynamics of perishability, elasticity, ageing of inventory, and markdowns. Their shortfalls, such as rigidity, stationarity and impossibility to scale up to real time supermarket dynamics, are an incentive to learning-based techniques that can dynamically adapt policies to the observed results.

2.2 Learning-Based Control (RL/DRL) for Dynamic Pricing & Ordering

Reinforcement learning offers an analytical alternative to analytical optimization based on learning pricing policies in response to repeated interaction with an environment. The application of RL is especially attractive in a perishable environment as the system will be path-dependent: the decision made today will impact on the subsequent remaining shelf life, the sell-through, and probability of spoilage. This renders closed-form optimization intractable on anything other than stylized or small-scale models.

The initial studies use RL in competitive and multi-agent pricing. As demonstrated by Kastius and Schlosser (2021), RL-based pricing may be superior to rule-based heuristics when it comes to competitive settings but points out that stability mechanisms are required; too much exploration may lead to unstable pricing. It underlines the importance of regulated action space and reward shaping, which are both implemented in the present project.

An increased amount of research in DRL subjects neural actor-critic and value-based approaches towards perishables. Yavuz and Kaya (2024) compare a set of DRL algorithms, such as DQN and PPO, and demonstrate that RL may be able to adapt to hard perishability, non-stationary demand, but these results are highly sensitive to the quality of the simulator and to the choice of hyperparameters. This justifies the use of a repeatable simplified environment such as it was done here.

Some DRL research incorporates a perishability aspect in the wider supply-chain choices. The approach to DRL in vendor-managed inventory systems of perishable goods is suggested by Mohamadi et al. (2024), where the authors present the efficiency of the learning-based policies in the simultaneous management of the upstream and downstream constraints. Zhou, Yang and Fu (2022) also add reference-price effects to their DRL model, which explains how the consumer memory of the path-dependent makes markdown decisions in two or more periods more complicated.

Other studies generalize RL to be behavioral or strategic-customer conditions. In Alamdar and Seifi (2024), strategic waiting of markdowns by customers comes into the limelight of importance of price-timing policies. Nomura, Liu and Nishi (2025) model perishability through age profiles and pipeline inventory that can be more accurate in determining the end-of-life price. Kavoosi et al. (2025) develop a combined RL pricing-inventory model in high-dimensional context and demonstrate apparent benefits in waste minimization and optimization of revenue.

Relevance to this study:

Despite the high potential of RL and DRL models, numerous previous studies are based on complex behavioral assumptions, rich state space or joint-pricing-ordering decisions. In comparison, the proposed project targets an open-air, lightweight RL environment that isolates fundamental perishability dynamics, namely stochastic demand, FEFO ageing and waste penalties, to determine whether a simple PPO policy can be more effective than more intuitively based supermarket baselines (Fixed and Random). This methodology focuses on reproducibility, interpretability and operational relevance instead of making novel discoveries that are algorithmic in nature.

2.3 Research Gap and Contribution

The review of the literature in Sections 2.1 and 2.2 demonstrates that both classical models and RL-based models can be of much help in explaining the pricing of perishable goods. The classical models provide a structural understanding of the demand elasticity, perishability, issuance rules and multi-stage shelf-life effects (Ding and Peng, 2024; Fan et al., 2020; Moshtagh et al., 2025; Rios and Vera, 2023; Solari et al., 2024; Vahdani and Sazvar, 2022). In contrast, RL and DRA models are flexible to stochastic demand, path-dependent inventory processes and multifaceted pricing-inventory relations (Kastius and Schlosser, 2021; Kavoosi et al., 2025; Mohamadi et al., 2024; Nomura et al., 2025; Alamdar and Seifi, 2024; Yavuz and Kaya, 2024; Zhou et al., 2022). Nonetheless, even with this extensive literature, there are still a number of gaps as compared to practical supermarket decision-making.

First, the vast majority of DRL studies use complicated and high dimensional simulators or have detailed behavioral assumptions (e.g. strategic customers, reference prices, social learning). The models are informative in theory but hard to reproduce and apply to working conditions where there is limited state information and interpretability is needed. More straightforward, open-source simulators which separate core perishability behavior, but which still can be used to benchmark the RL policies are needed.

Second, clear waste punishment and sustainability-oriented goals are not necessarily in the focal point in previous RL research. Numerous works optimize revenue or cost but pay little attention to spoilage as an optimization goal. In contemporary retailing, where one of the most important ESG concerns is the minimization of food-waste, it is worth having RL models that incorporate waste as a component of the reward signal and explicitly examine the trade-off between revenue and waste.

Third, when comparing various deep-learning algorithms RL studies do not commonly compare learned policies to what a practitioner would do in a real supermarket, e.g. fixed-price policies or naive non-learning policies. These baselines depict the current pricing of perishables as practiced by a number of retailers, and as such, the ability to determine whether RL can hold any significant value in outperforming them is crucial in determining the actual application.

This paper fills these gaps by using a lightweight, reproducible, expiry-conscious simulation framework that models the fundamental aspects of supermarket perishability the FEFO ageing of inventory, stochastic demand, discrete and explicit price multipliers and a penalty on out-of-date inventory. In this setting, we compare the performance of a PPO-based reinforcement learning policy against two intuitive baselines: Fixed and Random-based on the joint objective of maximizing revenue and minimizing waste.

The value of this work is thus three-fold:

1. An easy to understand and repeatable perishable goods simulator that attempts to isolate relevant mechanisms of perishability without unnecessary behavior complexity.
2. Strict comparative analysis of PPO versus realistic supermarket baselines, based on multi-seed experiments, analysis of temporal behavior and reward dispersion to measure robustness.
3. The empirical finding that an easy RL policy can internalize the constraint of perishability and can attain a more favorable revenue-wastes ratio than fixed or random pricing that can show how operational RL can be used in perishable pricing without the need to have large-scale and high-dimensional modelling.

On the whole, the project addresses a gap in theoretically rich DRL literature, and practical supermarket pricing demonstrating that a small RL system can create behavior that aligns with both waste reduction and revenue preservation, which are among the priorities of sustainable retail practice.

3 Research Methodology

The chapter outlines the entire process of research that was conducted in this study starting with the collection of data to the ultimate assessment of the pricing policies that were based on reinforcement learning. The emphasis is not on a generic analytics framework but on research methodology (what was done to produce and analyse results). CRISP-DM is informally followed in the general workflow (understanding of the data, prep, modelling and evaluation) but is not a formal structure here, and instead our focus is on the steps in the experiment that can be verified and can be replicated by other researchers.

3.1 Research Design and Overall Process

We formulate dynamic pricing of perishable goods as a series of decision-making and evaluate whether a reinforcement-learning (RL) policy can be used to increase the revenue-waste trade-off as compared to simple baselines. The steps are broken down into four stages:

1. Single store-product family data panel construction (daily granularity).
2. Simulation environment (Gymnasium) modelling ageing, demand, and spoilage inventory.
3. Policy training and selection; comparison of a learned PPO agent to two baselines (Fixed, Random).
4. Analysis and evaluation of 589-day episodes on various random seeds, which report revenue, waste, and reward.

3.2 Materials and Software

The experiments use a daily retail panel that is built in EDA_code.ipynb. The notebook reads multiple CSV files with historical store-level information (which is similar in structure with the Corporacion Favorita dataset) and combines it into a single DataFrame with a single row each day of the time a particular store and product family. Final panel contains fields such as date, baseline demand, promotion fraction, footfall, holiday flag, day of week, month and oil price.

Every experiment is coded in Python. Data preparation is done with pandas/NumPy, figures generation is done with matplotlib, the environment is defined with Gymnasium and RL is implemented with stable-baselines3 (PPO). The runs can be run on a regular laptop (8-16 GB RAM); no GPUs are necessary. Both environment and training are controlled through randomness using fixed seeds. Hyperparameters (e.g. price multipliers, elasticity, shelf life, waste penalty, number of episodes, learning rates) are written into recognizable configuration cells, and allow perfect replication of the experiment configuration.

3.3 Data Compilation, Sampling and Preparation

The sample is characterized as all the daily observations of a chosen store and a family of perishable products within a time frame of choice (e.g. a few months to a year). In EDA_code.ipynb, the notebook inserts raw transactional and calendar data (e.g. sales, store metadata, table of holidays, oil price histories), filter out irrelevant stores and products and aggregate transactions to daily level such that each row will represent one day of decision with total units sold, total transactions (to give footfall) and promotion flags. From these, calendar features (day of week, month, holiday indicator), a baseline demand series (base_demand), promotion intensity (promo_frac) and footfall proxy (daily transactions) were derived. The rows that lack critical values (base_demand, footfall, calendar fields) are dropped or imputed based on simple, conservative rules (e.g. drop days with no sales and no transactions). The demand here is tested to be generally more on high-footfall days and that promotions and holidays have the desired trends (using EDA plots and summary statistics), like those reported in perishable demand literature (Fan et al., 2020; Moshtagh et al., 2025; Solari et al., 2024; Yavuz and Kaya, 2024).

3.4 Environment and Scenario Set-Up

The custom PricingEnv models a single perishable family whose ageing is FEFO. At any given step (day) the agent chooses one of a finite number of price multipliers (e.g. 0.5, 0.7, 1.0, 1.1). The youngest bucket has some inventory added to it; the buckets get older each day; the oldest bucket goes out of stock. Stochastic demand is attracted with mean which is influenced by base demand and promotion (with implicit price elasticity by the multiplier). The revenue less a penalty on units that have expired can be used as a reward per step; horizon totals determine the episode-level revenue, waste, and reward. The policy requires the variables contained in the observation vector (e.g., base_demand, promo_frac, inventory age profile, last action).

3.5 Policies and Training Procedures

3.5.1 Baselines

- Fixed: will always take the base price multiplier (there will be no discount).
- Random: samples are taken equally through the set of actions.

3.5.2 RL Policy (PPO)

A Proximal Policy Optimization (PPO) agent is trained on the Stable-Baselines3 implementation on the same environment and reward function. PPO is an actor critic algorithm that optimizes a clipped surrogate goal that estimates advantages, stabilizing learning by ensuring the policy changes are not too large. A multilayer perceptron consisting of two hidden layers of 64 units and tanh activations is the representation of the policy and value functions are the default architecture used in Stable-Baselines3.

The PPO hyperparameters in this project are based mostly on the Stable-Baselines3 defaults but explicitly set up to be applicable to the short horizon supermarket setting. The agent is trained unless otherwise with: learning rate = $3 * 10^{-4}$, discount factor γ = 0.99, number of steps per rollout n_steps = 256, minibatch size = 256, GAE parameter λ = 0.95 (default), clipping range $|\varepsilon|$ = 0.2 for policy update (default), entropy coefficient = 0.01, value-function coefficient = 0.5 (default).

Shorter rollouts (n_steps = 256) and a larger batch size is then used to get a higher frequency of updating the parameters and also a smoother estimate of the gradients on this daily-control problem, and a small positive entropy coefficient (0.01) encourages sufficient exploration of price-multiplier space. The remaining parameters (λ, clipping range and value-function coefficient) will be left to their common default values.

The training is run to completion based on a fixed budget of timesteps, any number of passes over simulated episodes of 589 days, until the rolling mean episode reward reaches stabilisation. Exploration is turned off and the PPO agent is greedy in regard to the learned policy (with the use of mean action) during policy evaluation. To ensure that discrepancies in the total revenue, waste and reward can be related to the pricing policy, rather than randomness in the simulator, fixed, Random and PPO policies are always examined on the same environment setting, randomly selected seeds.

3.6 Measurements and Calculations

In the implementation, the anticipated demand on the day t is modelled as a Poisson rate which is a combination of the base demand, intensity of promotion and a selected price multiplier. In specific terms, the demand rate is

$$\boldsymbol{\lambda_t = \text{base\_demand}_t\,(1 + \beta_{promo}\ \text{promo\_frac}_t)\ a_t^{\varepsilon}}$$

where base_demand_t is the smoothed current day base demand, promo_frac_t is the promotion intensity, a_t is the policy choice of the discrete price multiplier (eg. 0.5, 0.7, 1.0, 1.1), β_{promo} is a promotion uplift coefficient and $\varepsilon < 0$ is the price elasticity parameter that causes the demand to decrease as price increases. The realized daily demand D_t is then drawn as

$$\boldsymbol{D_t \sim \text{Poisson}(\max(0, \lambda_t)).}$$

Daily revenue is calculated by multiplying the number of units sold and the effective unit price and the waste is the units that expire in the oldest inventory bucket. The learning scalar reward is

$$\boldsymbol{r_t = \text{revenue}_t - c_{waste} * \text{waste}_t,}$$

with $c_{waste} = 2$ in every observed experiment, such that the totals at the episode-level meet.

$$\textbf{TotalReward = TotalRevenue – 2 * TotalWaste}$$

This reward shaping explicitly represents the trade-off between revenue and waste and coincides with the results of the episode level shown in Tables 1 and 2.

3.7 Analysis and Validation

We run n= 5 seeds (each seed has 589 days of episode) and report:

- Average of seeds in each measure (policy ranking).
- Boxplot of the total reward by seeds (distributional robustness).
- Daily revenue time-series of qualitative operational behavior.

4 Design Specification

The architecture is designed in the form of a simulation-based reinforcement learning (RL) system that consists of three components:

- **Data preparation layer:** This layer is implemented in EDA_code.ipynb and it is used to build a daily panel of a given store-product family. Some of the panel variables (derived) that are needed include the baseline demand (base_demand), the intensity of promotion (promo_frac) and basic assumptions of inventory replenishment that will be needed in the simulator later. This panel is the sole input of the RL environment.

- **Simulation environment layer:** The custom PricingEnv was implemented in envpricing.py and it models the perishability and pricing problem as a Markov Decision Process. The environment exposes:
    - a state variable comprising of demand related variables the age-based inventory buckets and the prior price move,
    - a discrete space of price multipliers (e.g. 0.5, 0.7, 1.0, 1.1),
    - stochastic demand which is a Poisson model with mean determined by baseline demand and price multiplier of choice,
    - FEFO-based inventory ageing and expiry,
    - and a reward equal to daily revenue less a penalty on units which expire.
- **Learning and baseline layer:** This layer, implemented in rl_training_code.ipynb has:
    - Fixed-price baseline (always uses the base price multiplier),
    - Random baseline (acts in a uniform manner),
    - and a PPO agent that was trained with stable-baselines3.

    The PPO agent train policy which learns the state of the environment and maps this state to pricing actions so as to maximize the long-run reward (high revenue, minimum waste).

The simulated days go through the following loop: observe state → select a price multiplier → environment simulates demand, sales and expiry → reward returned → next state observed. PPO renews its policy parameters with fixed baselines.

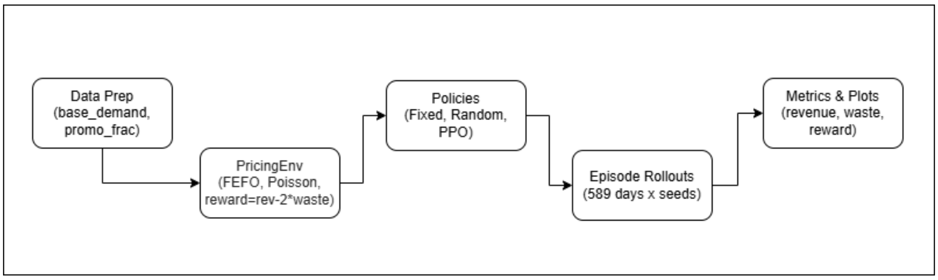

Fig 1. Architecture Diagram

5 Implementation

The chapter provides a summary of the code implementation of the data preparation, simulation environment, reinforcement learning policy and evaluation pipelines. Every element is coded in Python and is arranged into three primary artefacts: an exploratory notebook (EDA_code.ipynb), an environment module (envpricing.py) and a training/evaluation notebook (rl_training_code.ipynb).

5.1 Data and exploratory outputs

The notebook EDA_code.ipynb has the task of converting raw transactional and calendar tables into a daily panel that is ready to be ingested into the RL environment. It:

- filters the raw data to one target store and one family of perishable item

- combines item-level transactions into one row per day of aggregate units sold, transactions (an indicator of footfall) and a promotion flag
- adds calendar capabilities (day of week, month, holiday flag) and externals, including oil price
- makes a smooth base_demand series and a promo_frac feature which measures the intensity of promotions on a given day
- deletes or imputes rows that lack critical values (e.g. no sales and no transactions) in a conservative manner, which prevents the inclusion of unrealistic demand patterns.

The notebook also generates some of the important exploratory plots, such as skewed distribution of unit_sales to the right, time-series of daily demand of targeted family, and non-promotion and promotion groups summaries. These tests confirm that promotions and footfall are acting as predicted in literature and merit their representation in state representation of RL agent.

5.2 Environment module

The env_pricing.py file is a python implementation of the reusable PricingEnv class that follows the Gymnasium API (reset(), step()) and models FEFO ageing, replenishment, stochastic demand and reward computation. In each instance of the environment, there are:

- a fixed length array of inventory buckets that model the amount of the stock at each left age of shelf-life
- parameters of configuration like shelf life (number of buckets), daily replacement factor, maximum inventory and the discrete action set of price multipliers {0.5, 0.7, 1.0, 1.1}
- references to daily panel (base_demand, promo_frac and calendar variables) of the selected store-family.

The sequence of a single step(action) call is as follows:

- **Ageing and expiry:** The oldest bucket is taken off and handled as waste and rest of the buckets are moved one day towards expiry.
- **Replenishment:** The replenishment is the process that adds a new inventory to the young bucket, as per a replenishment rule that scales up the demand by a factor of safety-stock and the inventory is limited to a maximum level.
- **Demand simulation:** The predicted demand rate λ_t is calculated by using base_demand, promo_frac and selected action (price multiplier) and a Poisson random draw will generate the realized demand. The lowest demand and stock per bucket is the sales.
- **Revenue and reward:** The revenue is calculated as effective price times units sold; the waste is calculated as the amount of units in the oldest bucket that has expired. The reward $\mathbf{r_t = revenue_t - 2 * waste_t}$ is sent back to the agent along with the following state.

- **State construction:** The following observation concatenates base_demand, promo_frac, the entire set of inventory buckets and the prior action, which provides the agent with direct access to both demand drivers and the age profile of stock.

During an episode, the environment measures daily price multipliers, sales, waste and revenue and puts them into Python lists which are then transformed into pandas DataFrames, which are then used to plot and to compute episode-level totals. The NumPy and the environment are seeded with fixed random numbers so that it is possible to produce an experiment with the same precision.

5.3 PPO Training, Baselines and Evaluation

Training and evaluation occur with the help of the notebook rl_training_code.ipynb. It initially implements three policies on PricingEnv configuration:

- Fixed price policy, which always chooses the base price multiplier (action index corresponding to 1.0),
- Random policy, sampling uniformly over the discrete multiplier set,
- PPO agent, trained by Stable-Baselines3 on the hyperparameters in Section 3.5.2 using PPO("MlpPolicy", PricingEnv, etc.)

In the case of PPO, the notebook sets the environment to execute 589 days of consecutive episodes. Training occurs with a budget fixed to timesteps of several episodes till the rolling mean reward reaches some sort of steady state. It is then tested in the learned policy in exploitation mode with exploration noise off to get a deterministic path of prices in each test episode.

The assessment is structured into two situations:

- Single-episode comparison: both policies are executed on an identically initialized episode of 589 days and the totals of the revenue, waste and reward obtained are recorded (Table 1).
- Multi-seed robustness: the experiment is run with five random seeds; the results of the experiment are recorded; per policy, the code sums the total revenue, the total waste and total reward and then sums these three measures into seed-average measures (Table 2), a boxplot of total revenue across seeds and daily revenue time-series.

5.4 Tools and Reproducibility

All the parts are implemented in Python, using:

- pandas / NumPy for data manipulation,
- matplotlib to plot the demand distributions, the impact of promotion, revenue curves and reward boxplots,
- Gymnasium for the environment interface of PricingEnv,
- PPO training and evaluation using Stable-Baselines3.

The hyperparameters include price multipliers, elasticity coefficients, shelf life, waste penalty, PPO learning rate and the number of training timesteps, which are stated in distinctly identified configuration cells in the notebooks. This, combined with fixed random seeds, guarantees that everything in the Evaluation chapter can be re-run respectively on the same dataset and simulator configuration, and that everything will be exactly the same.

6 Evaluation

6.1 Experiment / Case Study 1 – Demand and Promotion Effects

This first experiment analyses the historical sales data for a single store–product family to understand basic demand patterns before training the RL agent. The filtered dataset contains 41,973 daily item-level records with five columns: date, store number, item number, unit_sales and a promotion flag. The distribution of unit_sales is right-skewed, with a mean of 5.34 units per record and a long tail up to 94 units, so values above the 99th percentile were clipped in the histogram to avoid outlier distortion.

Daily aggregation for the target store and family shows clear variability in total units sold over time, with visible peaks and troughs across the calendar. A simple grouping by promotion status indicates that average sales per record are higher on promotion days:

- Non-promotion days: 5.27 units per record
- Promotion days: 6.33 units per record

This difference justifies the promotion intensity, rolling demand characteristics, and new sales indicators in the state representation of the RL. These properties guarantee the ability of the pricing agent to base its decisions on observed shifts in demand.

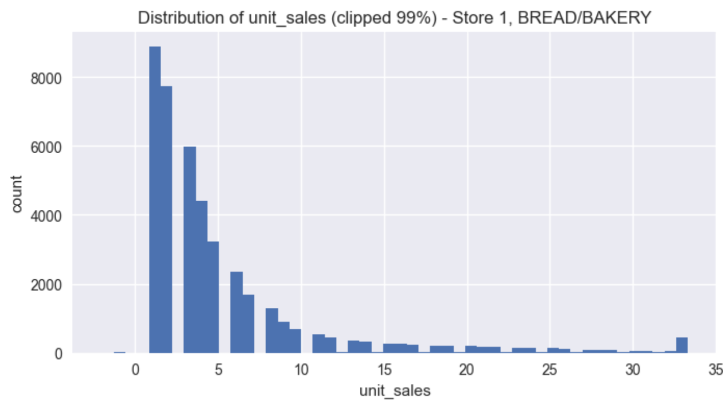

Fig 2. Distribution of unit_sales (clipped at 99th percentile)

6.2 Experiment / Case Study 2 – Single-Episode Policy Comparison

The second experiment evaluates three pricing policies on a single 589-day episode of the custom supermarket environment:

- Fixed price policy: always keeps the price multiplier at the base level (action = 2).
- Random policy: samples actions uniformly from the discrete action set.
- PPO policy: a trained Proximal Policy Optimization (actor–critic) agent from Stable-Baselines3, using the same environment and reward function.

For this single run, the following episode-level metrics were recorded:

Policy	Revenue	Waste (units)	Reward
Fixed	244636	45156	154324
Random	221537.6	11694	198149.6
PPO	203194.5	0	203194.5

Table 1. Metrics for single-episode policy comparison

The Fixed policy is the one with the most revenue though it experiences very high waste hence it is operationally infeasible with perishables. The PPO policy produces zero waste which proves that it learns to readjust price levels to clear ageing inventory. Its reward is best as the reward function is a direct penalty on unsold inventory on expiry. The Random policy lies in between the two baselines: moderate revenue, moderate waste. The theoretical assumption of this experiment is that the waste-sensitive reward function will prefer proactive inventory age management policies at the cost of marginal revenue. This is exactly what is observed in the experiment.

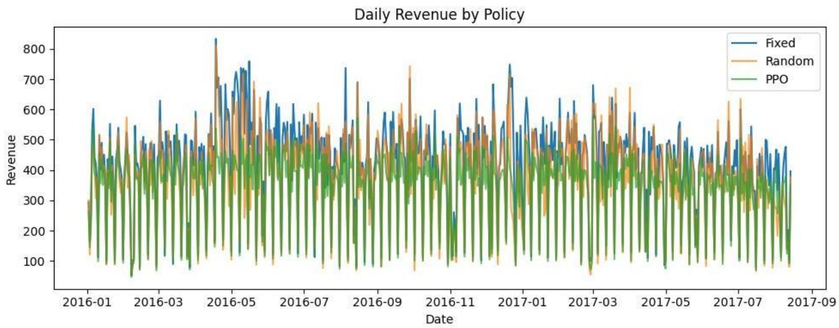

Fig 3. Daily Revenue Time Series for Fixed, Random and PPO Policies

6.3 Experiment / Case Study 3 – Multi-Seed Robustness Across Policies

In order to test whether the observed performance was not due to a single episode, the environment was tested in five random seeds.The average results are:

Policy	Total_Revenue	Total_Waste	Total_Reward
Fixed	246090.4	43703	158684.4
Random	218961.6	6773.8	205414
PPO	203182.9	0	203182.9

Table 2. Average Metrics Across Seeds (n=5)

The policy rankings are not only consistent across seeds, but are also strong: PPO is effective at zero waste and reward maximization. Fixed always generates too much waste that is not suitable. Random has mixed performance and is slightly inferior to PPO on reward.

To visualize the variability, Fig 3 shows us the reward distribution among seeds.

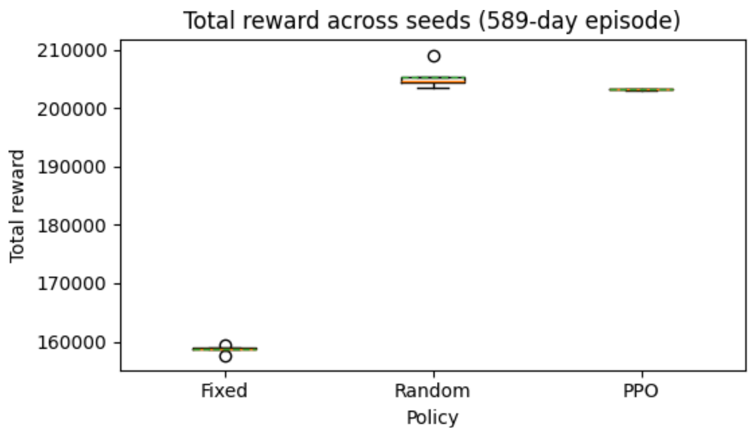

Fig 4. Boxplot of Total Reward Across Seeds (Fixed, Random, PPO)

The reward distribution in PPO is tightly clustered at the top of the range owing to zero waste at all times. Fixed has lowest and most scattered rewards and Random has moderate variability. The Fixed pricing method, still employed by many retailers, is proved to be operationally unsustainable when under the limitation of perishability because of excessive waste production. The assumption that stable pricing is a good approach to perishable inventory is thus falsified as a result of the experiment.

There were also negative findings: the Fixed-price baseline created the greatest amount of waste most of the time and was operationally inappropriate even with excellent revenue. Furthermore, there was a penalty-sensitivity effect: as the experiment was conducted with a low waste penalty (2x), the Random policy was sometimes slightly higher in reward than PPO on a few seeds. Nevertheless, the most consistent and sustainable policy was PPO and was the only one to reach zero waste. With stiffer penalties on waste, PPO would be likely to prevail in the reward and waste reduction.

The findings are clearly in line with the research hypothesis: Reinforcement-learning policy can be superior to traditional pricing heuristics because of its ability to manage both revenue and waste collectively, to have better operational results of perishable goods.

6.4 Discussion

The policy ranking is the same across 589 days of simulation, with five random seeds: Fixed maximizes revenue but generates extreme waste; PPO maximizes zero waste with the largest reward, other times the mean reward produced is slightly higher than that of PPO since the penalty is not very strong. The low variance of PPO (stable behavior), and moderate dispersion of Random are proved by the boxplot of total reward by seed, whereas the time series of daily revenue demonstrates that PPO is more stable in operations than Fixed/Random. The combination of the multi-seed averages, dispersion plot and the temporal traces give mutually reinforcing evidence that the learnt PPO policy is consistent in internalizing perishability. With that said, statistical power is low (n=5 seeds). We never used formal test of hypothesis or confidence interval on the inter-policy difference. Therefore, the conclusions are strong, but not inferential.

Scope of work - The environment characterizes an individual store-product family with discrete prices multipliers, decisions taken per day, shelf life that is finite and a reward that rewards a linear penalty on waste proportional to the quantity of units. The comparison is done on three policies (Fixed, Random, PPO) on episode totals (revenue, waste, reward) and their temporal patterns. Pre-RL EDA determines right-skewed demand, calendar effect, and quantifiable promotion effect; this is used to build the state representation of PPO.

Generalizability - The results most accurately predict those with a short shelf life or perishables that have daily pricing opportunities in which (i) relative price and promotion intensity are monotonically related to demand, (ii) inventory ageing is the dominant cost factor. Weaknesses in generalizability include: fines on waste are not linear (e.g. regulatory fines, reputational shocks), demand cross-elasticities are high (SKU substitution), delivery constraints, lead times or stockouts are binding, price is continuous and not discrete, high seasonality or exogenous shocks push the demand further than what the existing features pick up.

Strengths

Zero-waste control of PPO and low variance of seeds is an operationally meaningful finding.

Negative outcome is explicit: The high revenue of Fixed is not sustainable because it is extremely wasteful.

Transparency in design: the definition of rewards, penalty establishment and episode accounting are easily understandable.

Checks of robustness: multi-seed averages + distributional view (boxplot) + time traces.

Limitations and self-critique

Penalty sensitivity: the Random policy can be very close to or even surpassing PPO on reward with a rather small waste penalty (2x). An increased penalty will tend to turn the ordering decisively; therefore, conclusions are based on cost modelling.

Single-family level: there is no cross-SKU replacement, no category cannibalization, no lead-time stock replacement.

Action granularity: The discrete steps in price can be below or above optimal elastic responses; the continuous control can be discovered to be more effective in frontiers.

Evaluation horizon and inference: stability evidence can be found in 5 seeds, which have few statistical inferences; none of the bootstrapped CI or significance tests are reported.

Data realism: despite the utilization of demand and promotion indicators, the simulator does not consider marketing calendars, competitor pricing or weather shocks that have an impact in reality.

Relation to prior work - These findings are consistent with the literature which considers and treats perishability as a cost-of-waste control problem: reward shaping which prices inventory to clear before expiry has been shown to generally perform better than conventional heuristics. The trade-off curve of increased revenue and increased waste and the sensitivity of the optimum to the penalty are familiar in the control of inventory as well as cost-sensitive RL, and our experiments replicate these trends in a simplified supermarket model.

7 Conclusion and Future Work

The project investigated whether a reinforcement-based learning pricing policy could be more successful than the simple heuristics of perishable products at managing the revenues and waste together using a cost-conscious reward. In the given scope, the experiments demonstrate that a PPO-based agent is successful in achieving this goal on a regular basis: it removes waste not only among seeds but also maximizes the profit (reward) in the presence of penalties, that is, the learned policy takes into consideration perishability constraints better than fixed or random pricing.

The analysis of evidence is consistent. PPO achieves zero waste with low inter-seed variance, and its operations are more stable on daily operations whereas the fixed-price baseline, although it results in the highest top-line revenue, generates excessive waste that reduces effective profit and makes it operationally inappropriate. Reward competitiveness with the random policy only seems to be competitive with weak waste penalty; when the waste costs are more realistic or with higher costs, PPO is likely to prevail on both reward and sustainability. In general, the findings confirm the conclusion that reinforcing learning is capable of providing high-quality, consistent, and long term pricing choices in a perishable environment.

In the future, a few extensions would make it more realistic and higher quality of decision. Learning or establishing realistic costs of waste and tracing the revenue-waste Pareto frontier should be accomplished by penalty calibration and multi-objective RL. Cross-elasticities, competitor pricing, holidays and weather, promotion calendars and continuous pricing action can be added to demand modelling. There should be integration of operational constraints in terms of replenishment lead times, stockouts, service levels, and price-change budgets. To enhance inference, the studies should in future consider more seeds/episodes, confidence intervals and effect sizes, and ablation studies of features and action sets. Offline policy evaluation (e.g., counterfactual estimators) and staged A/B pilots are suggested to be used in deployment readiness. On the business side, a price-as-a-service layer that consumes the demand, inventory age, and promotion indicators, generates guarded price advice, and displays penalty sliders or visualization of the policy-frontier would enable the merchandisers to tune sustainability/profit on a category basis before it goes live - clear value to retailers operating under ESG and tight margin constraints would be provided.

References

Ding, J. and Peng, Z. (2024). Heuristics for perishable inventory systems under mixture issuance policies. Omega, [online] 126, p.103078. doi:https://doi.org/10.1016/j.omega.2024.103078

Fan, T., Xu, C. and Tao, F. (2020). Dynamic pricing and replenishment policy for fresh produce. Computers & Industrial Engineering, 139, p.106127. doi:https://doi.org/10.1016/j.cie.2019.106127

Kastius, A. and Schlosser, R. (2021). Dynamic pricing under competition using reinforcement learning. Journal of Revenue and Pricing Management, 21, pp.50–63. doi:https://doi.org/10.1057/s41272-021-00285-3

Kavoosi, A., Tavakkoli-Moghaddam, R., Sajedi, H., Tajik, N. and Tafakkori, K. (2025). Dynamic pricing and inventory control of perishable products by a deep reinforcement learning algorithm. Expert Systems with Applications, 291, p.128570. doi:https://doi.org/10.1016/j.eswa.2025.128570

Mohamadi, N., Niaki, S.T.A., Taher, M. and Shavandi, A. (2024). An application of deep reinforcement learning and vendor-managed inventory in perishable supply chain management. Engineering Applications of Artificial Intelligence, [online] 127, p.107403. doi:https://doi.org/10.1016/j.engappai.2023.107403

Moshtagh, M.S., Zhou, Y. and Verma, M. (2025). Dynamic inventory and pricing control of a perishable product with multiple shelf life phases. Transportation Research Part E Logistics and Transportation Review, 195, pp.103960–103960. doi:https://doi.org/10.1016/j.tre.2025.103960

Nomura, Y., Liu, Z. and Nishi, T. (2025). Deep Reinforcement Learning for Dynamic Pricing and Ordering Policies in Perishable Inventory Management. Applied Sciences, 15(5), pp.2421–2421. doi:https://doi.org/10.3390/app15052421

Parisa Famil Alamdar and Abbas Seifi (2024). A deep Q-learning approach to optimize ordering and dynamic pricing decisions in the presence of strategic customers. International Journal of Production Economics, pp.109154–109154. doi:https://doi.org/10.1016/j.ijpe.2024.109154

Rios, J.H. and Vera, J.R. (2023). Dynamic pricing and inventory control for multiple products in a retail chain. Computers & Industrial Engineering, 177, p.109065. doi:https://doi.org/10.1016/j.cie.2023.109065

Solari, F., Lysova, N., Bocelli, M., Volpi, A. and Montanari, R. (2024). Perishable Product Inventory Management In The Case Of Discount Policies And Price-Sensitive Demand: Discrete Time Simulation And Sensitivity Analysis. Procedia Computer Science, 232, pp.1233–1241. doi:https://doi.org/10.1016/j.procs.2024.01.121

Vahdani, M. and Sazvar, Z. (2022). Coordinated Inventory Control and Pricing Policies for Online Retailers with Perishable Products in the Presence of Social Learning. Computers & Industrial Engineering, 168, p.108093. doi:https://doi.org/10.1016/j.cie.2022.108093

Yavuz, T. and Kaya, O. (2024). Deep reinforcement learning algorithms for dynamic pricing and inventory management of perishable products. Applied Soft Computing, 163, p.111864.

doi:https://doi.org/10.1016/j.asoc.2024.111864

Zhou, Q., Yang, Y. and Fu, S. (2022). Deep reinforcement learning approach for solving joint pricing and inventory problem with reference price effects. Expert Systems with Applications, 195, p.116564.
doi:https://doi.org/10.1016/j.eswa.2022.116564

Dataset Link: https://www.kaggle.com/datasets/ruiyuanfan/corporacin-favorita-grocery-sales-forecasting